

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Control of Magnetic Bearing System

Hwang Hun Jeong, So Nam Yun and Joo Ho Yang

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/51185>

1. Introduction

A bearing fixes a rotating spindle to a specific location and is a mechanical component that supports the load applied to the axis and its dead load. Therefore, it is inevitable for mechanical contact between the axis and the bearing to occur, causing friction, abrasion, heat, noise, and user environment contamination from lubrication. Magnetic bearings are mechanical components that use the attractive or repulsive force of electromagnets to support the mechanical axis in a non-contact state. The use of such components significantly reduced the disadvantages that accompany the use of general mechanical bearings such as friction, abrasion, heat, noise, and user environment contamination from lubrication. Moreover, magnetic bearings can support the mechanical axis in special environments (vacuum, high temperature, low temperature, zero gravity) and have the advantage of being able to adjust the damping coefficient and spring constant of the system that supports the axis according to the control objective.

Magnetic levitation can be categorized into the following systems depending on the form of force that supports the levitated object: the system that uses magnetic attraction, magnetic repulsion, induction levitation, and superconducting Meissner Effect. Magnetic levitation that utilizes attractive force has a closed magnetic circuit so efficiency is high and 1-axis control is possible due to the stability in the attraction and perpendicular directions. However, it has been reported that the uncontrolled directions have poor stability due to the nonlinearity of the attraction. Magnetic levitation that uses the repulsive power has stable characteristics with respect to the longitudinal direction that the repulsive force is applied to, but the transverse direction has unstable characteristics. However the electromagnet is arranged, all the axes cannot be stabilized. Magnetic levitation that uses induction levitation is able to perform stable levitation without special control as Fleming force caused by the relative velocities between the electromagnet and the conductor supports the levitation. However, without a velocity over a certain level, levitation cannot be supported where overall efficiency is low due to Eddy current loss

and brake loss. Magnetic levitation that uses superconducting Meissner Effect takes advantage of the repulsion with permanent magnets caused by the strong diamagnetism from the superconductor. Like that of the induction levitation, stable levitation is possible without any control. However, the operational temperature of the levitation system using superconductor is very low: 4.2K(liquid Helium), 77K(liquid Nitrogen). Generally, the magnetic levitation applied to the magnetic bearing is the method using attraction and repulsion. Magnetic bearing systems discussed here refers to a system that utilizes the attraction.

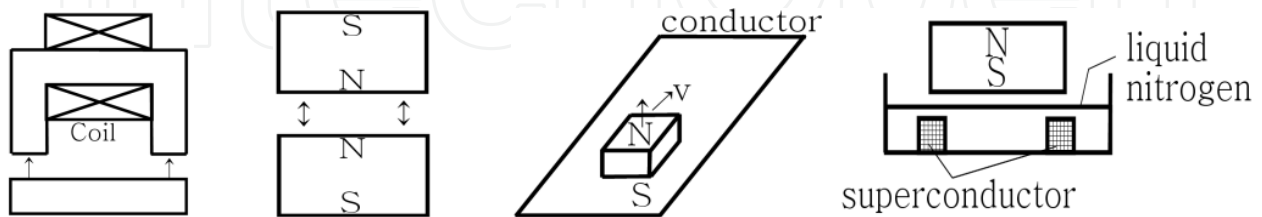


Figure 1. Levitation system according to the magnetic levitation method

In this chapter, the method to designing a magnetic bearing system, to obtaining a mathematical model, and understanding the preparations necessary for control will be discussed. For this, the calculation of attraction using the Probable Flux Paths Method, selection of circuitry about the amplifier to operate the electromagnet, and method to identify the magnetic bearing system that includes a PID controller are discussed.

2. Magnetic bearing system design

Fig. 2 shows a schematic diagram of the magnetic bearing system to be designed in this chapter. The levitated object is supported by the attraction of the electromagnet and the attraction of the electromagnet is controlled by the current in the coil. In order to design and control such a magnetic bearing system, the amplifier to operate the electromagnet that composes the magnetic bearing system and hardware to control the whole system need to be designed first. Next, the designed magnetic bearing system is modeled mathematically, then the parameter values difficult to measure through the mathematical model are determined through experimentation. Finally, an adequate controller is designed and applied to the identified magnetic bearing system. In this chapter, the detailed control laws for magnetic bearing control are excluded and the implementation of the magnetic bearing system before applying various controllers is the main objective.

2.1. Magnetic bearing system composition

A magnetic bearing system like that of Fig. 2 is composed of the object to be levitated, core, electromagnet including the coil, amplifier to operate the electromagnet, displacement measurement system to measure the distance between the levitating object and the electromagnet, control law to calculate the control signal from the feedback signal, and control system that includes the hardware to realize the control law.

Fig. 3 is the assembly of the magnetic bearing system to make. In the assembly, the levitated object will be supported by magnetic bearing at X and Y axis direction. But thrust direction has only the mechanical backup bearing. And Fig. 4 is the levitated object that is 1.4kg.

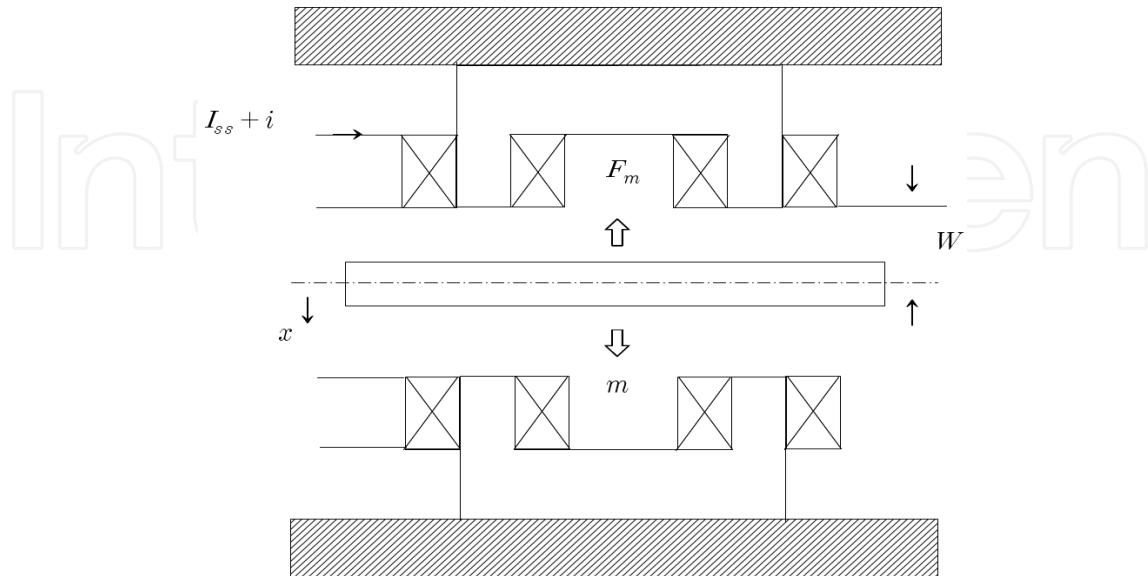


Figure 2. Schematic of the magnetic bearing system

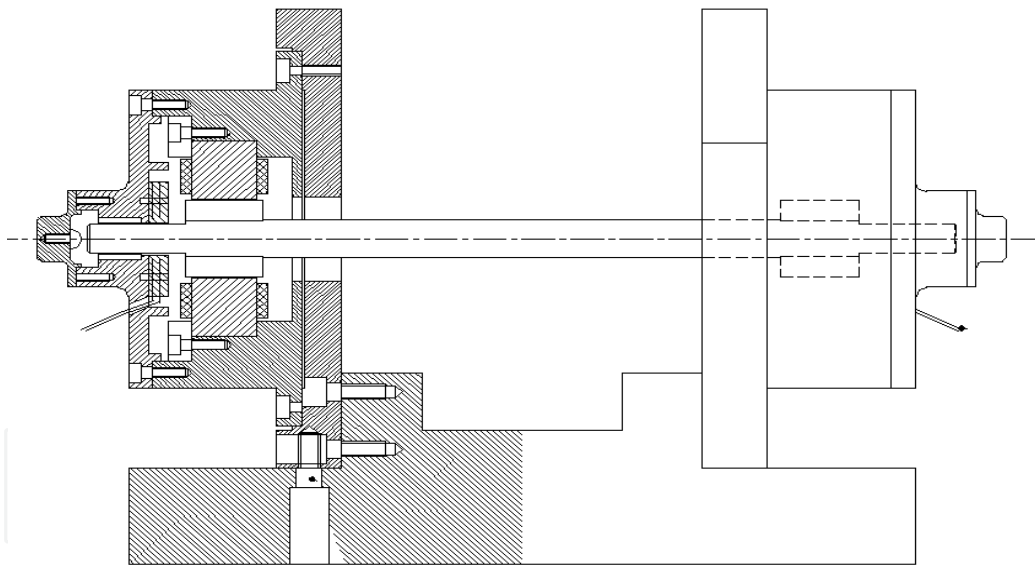


Figure 3. Assembly of the magnetic bearing system

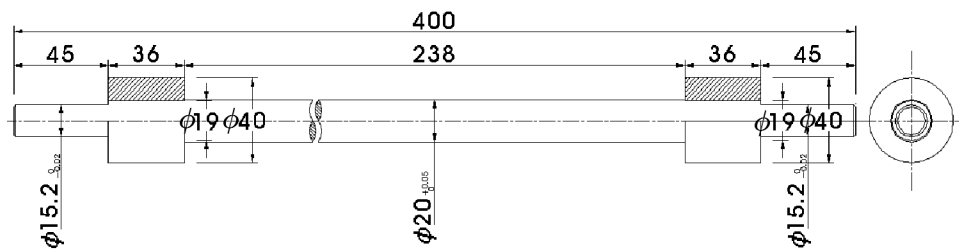


Figure 4. Levitated object

Since a magnetic bearing system like Fig. 3 has a symmetric form vertically and horizontally, the levitating object can be simply assumed as a point mass in the perspective that the object is levitated. Therefore, in this section, the elements of the magnetic bearing system excluding the levitating object are designed.

2.1.1. Probable flux paths method

To support the levitating object through the electromagnet's attractive force, the attraction relationship between the current in the electromagnet coil and the levitating object needs to be defined clearly. The Probable Flux Paths Method assumes that the magnetic permeability of the magnetic substance that forms the magnetic path is linear to calculate the permeance of the magnetic substance that the magnetic path passes through, followed by the calculations of the magnetomotive force, magnetic flux, magnetic flux density, and attractive force. As the permeability of materials disregarding permanent magnets are generally nonlinear, the error between the Probable Flux Paths Method calculation results and that of actual experimentation measurements is large and calculations by applying the Probable Flux Paths Method for magnetic substances with complicated magnetic paths is known to be difficult. However, since the vertically and horizontally symmetric magnetic bearing system magnetic path is of a simple form, the electromagnet is designed by applying the Probable Flux Paths Method early in the design process. For more precise designing, the use of FEM software such as Maxwell is desirable.

Generally, the following assumptions have to be satisfied when using the Probable Flux Paths Method to analyze the magnetic circuit.

- a. The relationship between magnetic flux and current is linear.
- b. The average magnetic flux passes through the centroid of the cross section.
- c. When the cross section that the magnetic flux passes through changes, the parts are calculated by dividing them into different parts and setting as combinations of parallel or series.
- d. When the cross section of a part changes rapidly, the magnetic flux passes through in a smooth circular arc (quadratic curve).

2.1.2. Electromagnet design

In order to design the electromagnet using the Probable Flux Paths Method, first, the attractive force derived from the magnetic circuit caused by the electromagnet needs to withstand the weight of the mass. Here, the following steps in design are taken so that sufficient attractive force from the electromagnet is produced for control.

- a. The mass of the levitating object is determined.
- b. The material of the core and levitating object is determined.
- c. The attractive force of the electromagnet is calculated with values determined by assumptions regarding the current, magnetic circuit, and length of the coil and number of windings during normal conditions.

- d. The material of the core and levitating object, coil and number of windings, and current at normal state is adjusted until the comparison of the calculated attractive force value and the mass of the levitating object gives a satisfactory attractive force.
- e. The electromagnet's maximum attractive force and coil's maximum current according to the core and saturation flux density of the levitating object material are calculated.
- f. The material of the core and levitating object, coil and number of windings, and current at normal state is adjusted until the calculated maximum attractive force of the electromagnet is sufficient.

Since the attractive force calculated through the Probable Flux Paths Method has a large error with the actual experimentation values, in order to manufacture magnetic bearings based on this design, it is desirable to design with a safety factor of greater than 3.

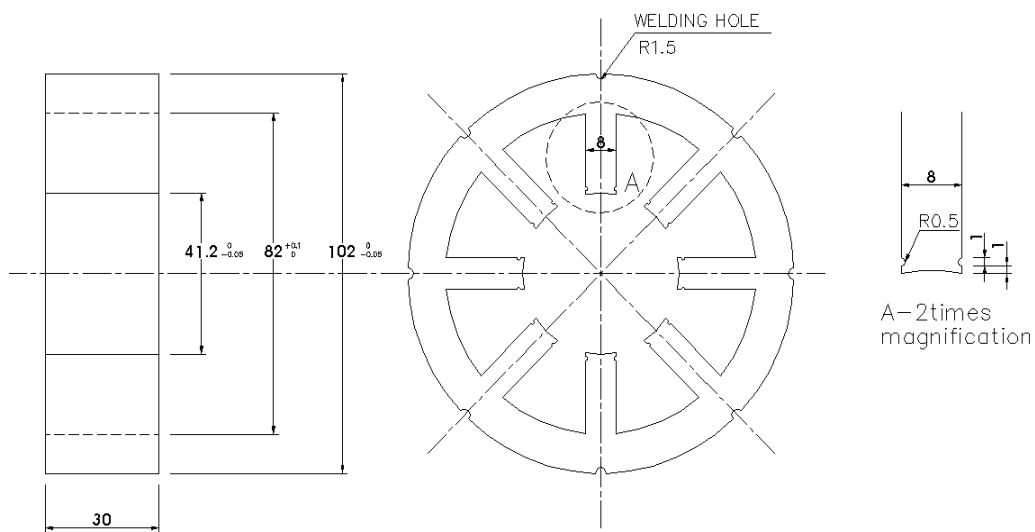


Figure 5. Electromagnet core

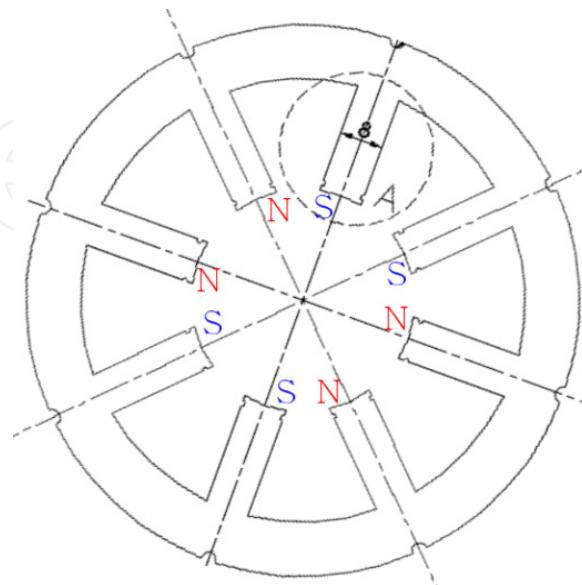


Figure 6. Electromagnet magnetic circuit formation

Next, the number of poles and the angle of the core have to be kept in mind. Fig. 5 shows the core to be used as the electromagnet in the magnetic bearing system. The magnetic circuit caused by the electromagnet that supports one axis has to be designed so that it does not interfere with the magnetic circuit caused by the electromagnetic that supports the other axis. The direction of the magnetic circuit caused by the electromagnet is determined by the direction of the coil wound about the core. The winding has to have polarity as shown in Fig. 6 so that interference of the magnetic circuit does not occur. Also, for the convenience of control, the resultant force of the attraction caused by the magnetic path needs to be parallel or perpendicular to the supporting axis.

2.1.3. Displacement measurement system design

When the current flowing in the coil is constant, the attractive force is proportional to the square of the distance to the levitating object. Therefore, in order to implement a magnetic bearing control system, the distance between the electromagnet and the levitating object has to undergo feedback.

Sensors that measure gaps in noncontact state include using the change in capacitance, change in Eddy Current, and using laser or ultrasound. A displacement measurement sensor has to be selected with consideration of the sampling time, range and area of the gap to be measured, and economic feasibility of the overall system.

2.1.4. Control system design

The objective of a magnetic bearing control system is producing a control signal from the error signal between the reference input and the gap (between the electromagnet and the levitating object). And from this control signal, control the current of the electromagnet coil to reach the reference input in a stable manner. The control period of a control system is the time consumed in performing one-iteration of computations given by the control system. In order to implement a magnetic bearing system of high speed rotations, the control period has to be as small as possible. To implement such a system, the system is designed taking into consideration all the speeds of the MPU which will operate processing the response speed of the displacement sensor, A/D converter speed, D/A converter speed, and discretize the control laws.

2.2. Magnetic bearing system mathematical modeling

2.2.1. Relation between electromagnet and levitating object

First, the mathematical model for the case of the above existing electromagnet supporting the levitating object is derived. The equation of translational motion for the levitating object is as shown in Equation (1).

$$m \frac{dv}{dt} = mg - F_m \quad (1)$$

Here, m is the levitating object mass, g is the gravitational acceleration and F_m is the attractive force of the electromagnet.

Generally, the electromagnet's attractive force is as shown in Equation (2).

$$F_m = \frac{B^2}{2\mu_0} A_m \quad (2)$$

Here, B is the magnetic flux density, μ_0 is the relative permeability of the vacuum and A_m is the opposing area of the electromagnet. Thus, the attractive force of the electromagnet is determined by the opposing area of the electromagnet and the magnetic flux density.

The magnetic circuit and magnetic flux by the electromagnet is as shown in Equation (3).

$$\Phi = \frac{F_{mf}}{R_m} = \frac{NI_m}{\frac{l_m}{\mu_s}} = \frac{NI_m}{\frac{l_m}{\mu_0\mu_s S} + 2\frac{x_0}{\mu_0 S}} = \frac{\mu_0 S NI_m}{\frac{l_m}{\mu_s} + 2x_0} \quad (3)$$

Here, F_{mf} is the magnetomotive force, N is the number of coil windings, R_m is the magnetic resistance, l_m is the length of the magnetic path, μ is the relative permeability, x_0 is the gap between the electromagnet and the levitating object, μ_s is the relative permeability of the metal pin and S is the cross sectional area of the metal pin.

The magnetic flux density of the electromagnet is the magnetic flux per unit area and is as shown in Equation (4).

$$B = \frac{\Phi}{S} = \frac{\mu_0 NI_m}{\frac{l_m}{\mu_s} + 2x_0} \quad (4)$$

Therefore, the attractive force of the electromagnet is as shown in Equation (5).

$$F_m = \frac{\mu_0 N^2 I_m^2 S}{\left(\frac{l_m}{\mu_s} + 2x_0\right)^2} \quad (5)$$

If the electromagnetic coil current in neutral state is assumed as I_{ss} and the varying control current signal from the neutral state is assumed as i , Equations (5) and (6) can be put together and Equation (7) is satisfied.

$$F_m = k \left(\frac{I_{ss} + i}{X_0 + W + x} \right)^2 \quad (6)$$

$$mg - k \left(\frac{I_{ss}}{X_0 + W} \right)^2 = 0 \quad (7)$$

Here,

$$X_0 = \frac{I_{ss}}{2\mu_s} \text{ and } k = \frac{N^2 \mu_0 S}{4} .$$

To apply linear control theory, the control subject also has to be a linear control system. However, Equation (1) includes a nonlinear term, thus, linear control theory cannot be applied. So, the nonlinear term of Equation (1) is linearized through Taylor series.

Assuming that the values after the second-order Taylor series terms are sufficiently small compared to the first-order term, the Taylor series in the parallel point ($i = 0, x = 0$) with regard to the nonlinear term $k\left(\frac{I_{ss}+i}{X_0+W+x}\right)^2$ is as shown in Equation (8).

$$k\left(\frac{I_{ss}+i}{X_0+W+x}\right)^2 = k\left(\frac{I_{ss}+i}{X_0+W+x}\right)^2 \Big|_{i=0,x=0} + \frac{\partial}{\partial i} k\left(\frac{I_{ss}+i}{X_0+W+x}\right)^2 \Big|_{i=0,x=0} (i-0) + \frac{\partial}{\partial x} k\left(\frac{I_{ss}+i}{X_0+W+x}\right)^2 \Big|_{i=0,x=0} (x-0)$$

$$k\left(\frac{I_{ss}+i}{X_0+W+x}\right)^2 = k\left(\frac{I_{ss}}{X_0+W}\right)^2 + \frac{2kI_{ss}}{(X_0+W)^2} i - \frac{2kI_{ss}^2}{(X_0+W)^3} x \quad (8)$$

Equation (8) is substituted into Equation (1) to obtain Equation (9) the equation of translational motion of the levitating object.

$$m\ddot{x} = mg - \left\{ k\left(\frac{I_{ss}}{X_0+W}\right)^2 + \frac{2kI_{ss}}{(X_0+W)^2} i - \frac{2kI_{ss}^2}{(X_0+W)^3} x \right\} \quad (9)$$

Taking into consideration the conditions of Equation (7), Equations (9) can be rearranged into the following Equation (10).

$$m\ddot{x} = \frac{2kI_{ss}^2}{(X_0+W)^3} x - \frac{2kI_{ss}}{(X_0+W)^2} i \quad (10)$$

The relationship between the current flowing in the coil and the drop of electric pressure in the coil of the upper side electromagnetic in the magnetic bearing is shown as Equation (11).

$$\frac{d}{dt}(L(I_{ss} + i)) + R(I_{ss} + i) = E_{ss} + e \quad (11)$$

Here, E_{ss} is the voltage that appears due to the current flowing in the electromagnet coil in neutral state, e is the voltage that appears due to the control current flowing in the electromagnet coil, and Equation (12) is satisfied.

$$RI_{ss} = E_{ss} \quad (12)$$

Additionally, the inductance of the coil is proportional to the number of coil windings and the magnetic flux as shown in Equation (13).

$$L = \frac{N\Phi}{I_m} \quad (13)$$

When assuming that there is no leakage magnetic flux in the magnetic path caused by the coil, the coil inductance is as shown in Equation (14).

$$L = \frac{Q}{X_0+W+x} \quad (14)$$

Here, W is the gap between the electromagnet at neutral state and the levitating object and x is the gap between the electromagnet and levitating object varying due to the control input.

Moreover, $X_0 = \frac{I_{ss}}{2\mu_s}$ and $Q = \frac{N^2\mu_0 S}{2}$.

If L_0 is the leakage magnetic flux, the coil inductance is as shown in Equation (15).

$$L = \frac{Q}{x_0 + W + x} + L_0 \quad (15)$$

Therefore, the term $\frac{d}{dt}(L(I_{ss} + i))$ in Equation (11) can be solved like that of Equation (16), and Equation (11) is the same as Equation (17).

$$\begin{aligned} \frac{d}{dt}(L(I_{ss} + i)) &= L \frac{d}{dt}(I_{ss} + i) + (I_{ss} + i) \frac{dL}{dt} \\ \frac{d}{dt}(L(I_{ss} + i)) &= L \frac{di}{dt} + (I_{ss} + i) \frac{\partial L}{\partial x} \frac{dx}{dt} \\ \frac{d}{dt}(L(I_{ss} + i)) &= L \frac{di}{dt} + (I_{ss} + i) \frac{-Q}{(x_0 + W + x)^2} \frac{dx}{dt} \end{aligned} \quad (16)$$

$$L \frac{di}{dt} - \frac{Q}{(x_0 + W + x)^2} \dot{x}(I_{ss} + i) + R(I_{ss} + i) = E_{ss} + e \quad (17)$$

Equations (17) and (18) can be obtained from the conditions of Equation (12).

$$L \frac{di}{dt} - \frac{Q}{(x_0 + W + x)^2} \dot{x}(I_{ss} + i) + Ri = e \quad (18)$$

If the levitating object is assumed close to neutral state ($i = 0$, $x = 0$) and I_{ss} is sufficiently larger than i , Equation (18) is the same as Equation (19).

$$L \frac{di}{dt} - \frac{Q}{(x_0 + W)^2} \dot{x}I_{ss} + Ri = e \quad (19)$$

2.2.2. Relationship of linear amplifier

Fig. 7 shows the electric circuit of the amplifier to operate the electromagnet coil.

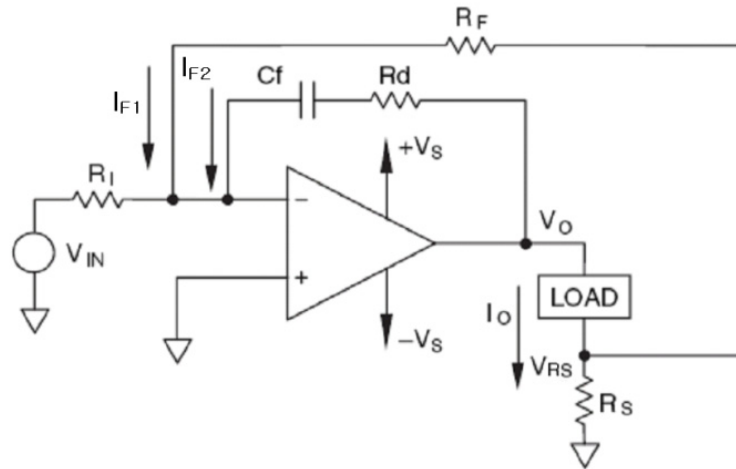


Figure 7. Current amplifier circuit

Assuming the current amplifier as an ideal amplifier, the transfer function from the amplifier control input V_{IN} to the load current I_o is found to be as shown in Equation (20).

$$-\frac{R_I}{R_F} R_S I_o - \frac{R_I}{Z_I} (Z_L + R_S) I_o = V_{IN}$$

$$\begin{aligned}
 -\left(\frac{R_1 R_S}{R_F} + \frac{R_1(Z_L + R_S)}{Z_1}\right) I_o &= V_{IN} \\
 -\left(\frac{R_1 R_S Z_1 + R_F R_1(Z_L + R_S)}{R_F Z_1}\right) I_o &= V_{IN} \\
 \frac{I_o}{V_{IN}} &= -\frac{R_F Z_1}{R_1 R_S Z_1 + R_F R_1(Z_L + R_S)} \tag{20}
 \end{aligned}$$

In this circuit, when the impedance Z_1 and load impedance Z_L undergoes Laplace Transformation for the transient characteristic improvement of the amplifier, Equations (21) and (22) are obtained.

$$Z_1 = \frac{R_d C_f s + 1}{C_f s} \tag{21}$$

$$Z_L = Ls + R \tag{22}$$

Substituting Equations (21) and (22) into Equation (20) and rearranging the equation, the transfer function from the amplifier control input V_{IN} to the load current I_o can be found like Equation (23).

$$\begin{aligned}
 \frac{I_o}{V_{IN}} &= -\frac{R_F \frac{R_d C_f s + 1}{C_f s}}{R_1 R_S \frac{R_d C_f s + 1}{C_f s} + R_F R_1 (Ls + R + R_S)} \\
 \frac{I_o}{V_{IN}} &= -\frac{R_F (R_d C_f s + 1)}{R_1 R_S (R_d C_f s + 1) + R_F R_1 (Ls + R + R_S) C_f s} \\
 \frac{I_o}{V_{IN}} &= -\frac{R_F R_d C_f s + R_F}{R_F R_1 L C_f s^2 + (R_1 R_S R_d + R_1 R_F (R + R_S)) C_f s + R_1 R_S} \tag{23}
 \end{aligned}$$

2.2.3. Block diagram and transfer function of the overall system

The block diagram of the magnetic bearing system using an upper electromagnet from the relationships of Equations (10), (19), and (23) is shown in Fig. 8.

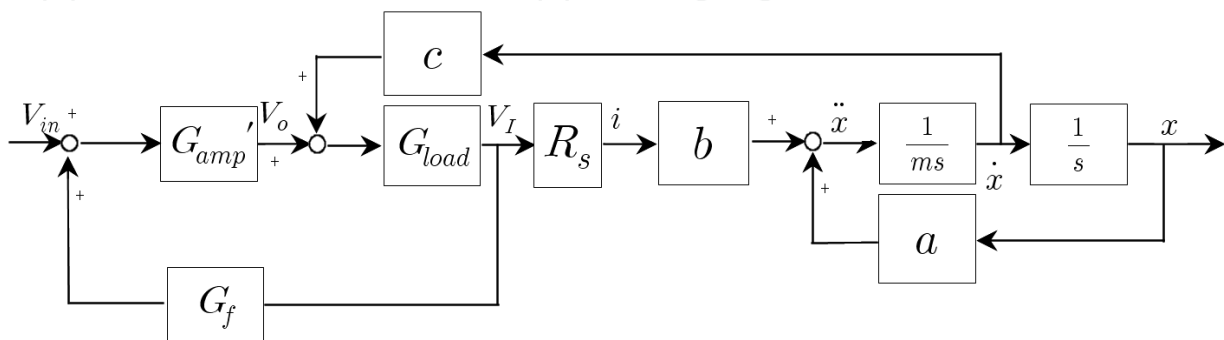


Figure 8. Block diagram of the magnetic bearing system

Here, the transfer function G_{MB} from the current amplifier circuit input voltage V_{IN} to the air gap x between the magnetic bearing and the levitating object is as shown by Equation (24).

$$G_{MB} = \frac{b_{mb1}s + b_{mbo}}{a_{mb4}s^4 + a_{mb3}s^3 + a_{mb2}s^2 + a_{mb1}s + a_{mbo}} \quad (24)$$

Here,

$$a_{mb4} = R_I R_F C_f L m$$

$$a_{mb3} = R_I R_F C_f m (R + R_s) + R_I R_d C_f m$$

$$a_{mb2} = R_I m - R_I R_F C_f (b c R_s + a L)$$

$$a_{mb1} = -a (R_I R_F C_f (R + R_s) + R_I R_d C_f)$$

$$a_{mb0} = -a R_I$$

$$b_{mb1} = -b R_F R_d C_f$$

$$b_{mb0} = -b R_F$$

$$a = \frac{2k_{ss}^2}{X_1^3}$$

$$b = -\frac{2k_{ss}}{X_1^2}$$

$$c = \frac{2k_{ss}}{X_1^2}$$

$$X_1 = X_0 + W + x$$

$$X_0 = \frac{l_m}{2\mu_s}$$

3. Magnetic bearing system control

To control the mathematically modeled magnetic bearing system, a process to design the driver to operate the electromagnet and a process to identify unknown parameters are necessary. Here, the method to determine the peripheral device values of the linear amplifier circuit that has the desired output by applying a genetic algorithm and the method to identify the magnetic bearing system using a PID controller to stabilize the genetic algorithm and system are discussed.

3.1. Genetic algorithm

In the method to design a linear amplifier with an output sought by the designer or a method to identify the system parameters through random experimentation data, there are methods available using a frequency response method and applying genetic algorithm. Here, the method to apply genetic algorithm and selecting the desired value is explored.

Genetic algorithm is an algorithm that imitates genetics and natural evolution to optimize the objective function and find the solution set with a structure as shown in Fig. 9.

Genetic algorithm initially generates an initial group to solve the optimization problem defined mathematically. Through the difference with the objective function, the degree of agreement of the chromosomes in the generated initial group is calculated and the result of the calculation becomes the basis for dividing the chromosomes into dominant and recessive chromosomes. Through the reproduction operation based on the degree of agreement of the initial group, they become the source of breeding and through crossover operation, a temporary population is generated. Mutation operation on the generated temporary population leads to the generation of the next generation population. The

process of generating the next generation population after going through the aforementioned series of processes is described as one generation and the method to finding an optimized solution to an objective function through operations in a specific generation is defined as an algorithm.

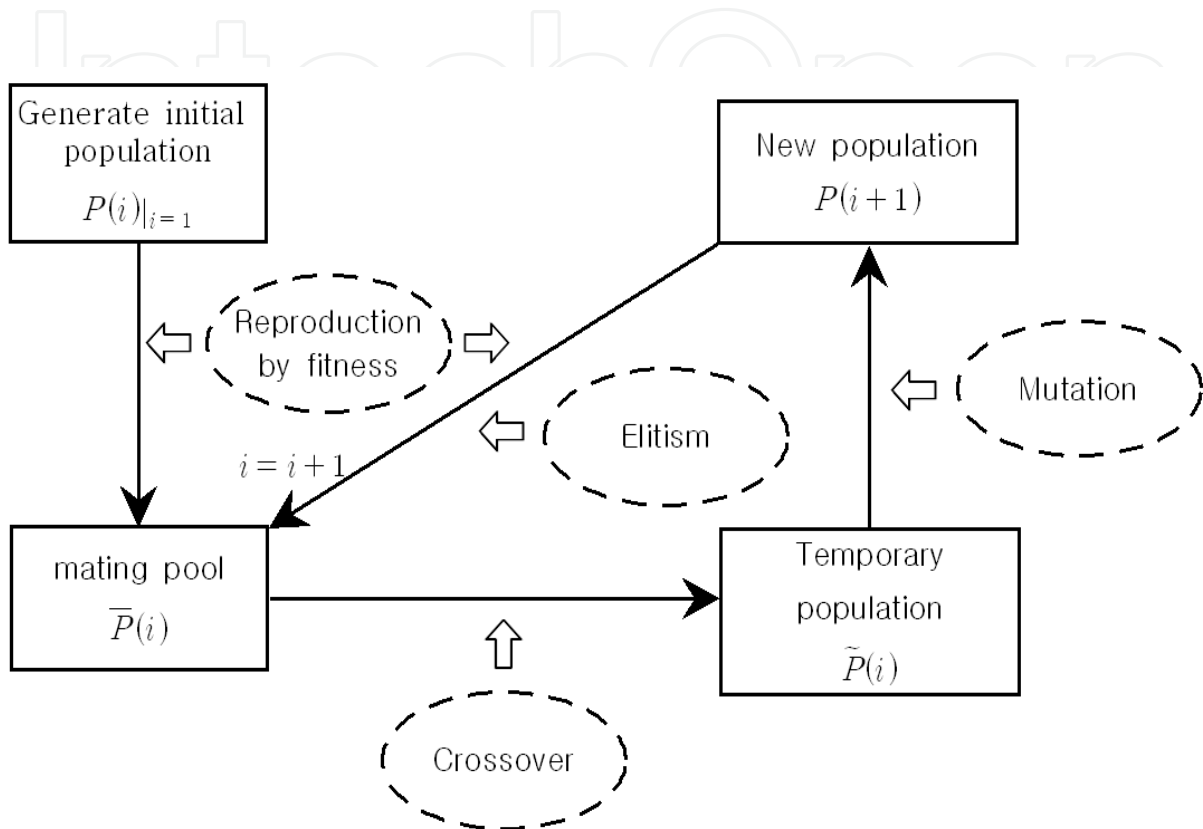


Figure 9. Genetic algorithm

Genetic algorithms can be categorized into BCGA(Binary Coded Genetic Algorithm), SGA(Signal Genetic Algorithm), and RCGA(Real Coded Genetic Algorithm) depending on the expression of the chromosome. Generally, RCGA is used for optimization problems regarding continuous search domain variable with constraints. This is because if the chromosome is expressed by real code, genes that match perfectly with the variable in question could be used and the degree of precision of the calculation is only dependent on the calculation ability of the computer regardless of the length of the gene.

3.2. Amplifier peripheral circuit design

As can be seen in Fig. 7, the linear amplifier circuit is composed of resistance R_1 which determines the amplification ratio of the current amplifier and the amplifier output current, resistance R_F , resistance R_d which determines the dynamic characteristics of the linear amplifier circuit, condenser C_f , R_s which limits the linear amplifier circuit current, and the

load(here, coil). When defining the form of output desired by the designer using time response characteristics, the amplifier peripheral device values can be determined in the following manner utilizing genetic algorithm.

First, the value of the part to solve is defined. Since the amplification ratio A of the current amplifier, current limiting resistance R_s , load inductance L , and resistance R are unknown, the variables of the genetic algorithm to find are limited to the resistance R_I that determines the amplification ratio of the amplifier output current, resistance R_F , resistance R_d which determines the dynamic characteristics of the linear amplifier current, and condenser C_f . At this point, if the amplification ratio of the amplifier output current is given, one less genetic algorithm variable needs to be found as the resistances R_I and R_F have a proportional relationship.

Next, the searching range of the parameters to be identified is limited according to the characteristics of each device. As the resistance R_I which determines the amplification ratio of the amplifier output current is a signal resistance, it is desirable to have a high resistance value. Therefore, in the case of resistance R_I which determines the amplification ratio of the amplifier output current, it has to be sought in the $k\Omega$ range. In contrast, resistance R_d which determines the dynamic characteristics of the linear amplifier circuit has to be sought in a wide range. For condenser C_f , which determines the dynamic characteristics of the linear amplifier circuit, a value in the nF to μF range is ideal when considering the dynamic characteristics of the current amplifier.

After that, the objective function is determined to implement the genetic algorithm. The objective in this program is the design of a linear current amplifier that has a current output in the form that the designer seeks. Therefore, it has the form shown in Equation (23) and the response of the system that satisfies the time response characteristics defined by the designer is defined as shown in Equation (25).

$$G_r(s) = -\frac{e_1 s + e_0}{s^2 + d_1 s + d_0} \quad (25)$$

At this point, the randomly given d_0 , d_1 , e_0 and e_1 are the coefficients of the system G_r that satisfies the time response characteristics. The objective function to implement the genetic algorithm is defined as shown in Equation (26).

$$F_{obj} = \int e(t)^2 dt \quad (26)$$

Here, $e(t) = g_r(t) - g_{amp}(t)$, $g_r(t)$ is the step response of the system defined by the designer and $g_{amp}(t)$ is the step response of the current amplifier transfer function.

Finally, the parameters to operate the genetic algorithm, such as the size of the entity group, the maximum chromosome length, maximum number of generations, crossbreeding probability, and mutation probability, are defined. Here, in order to improve the performance of the implemented genetic algorithm, configuration for methods such as the penalty strategy, elite strategy, and scale fitting method is necessary.

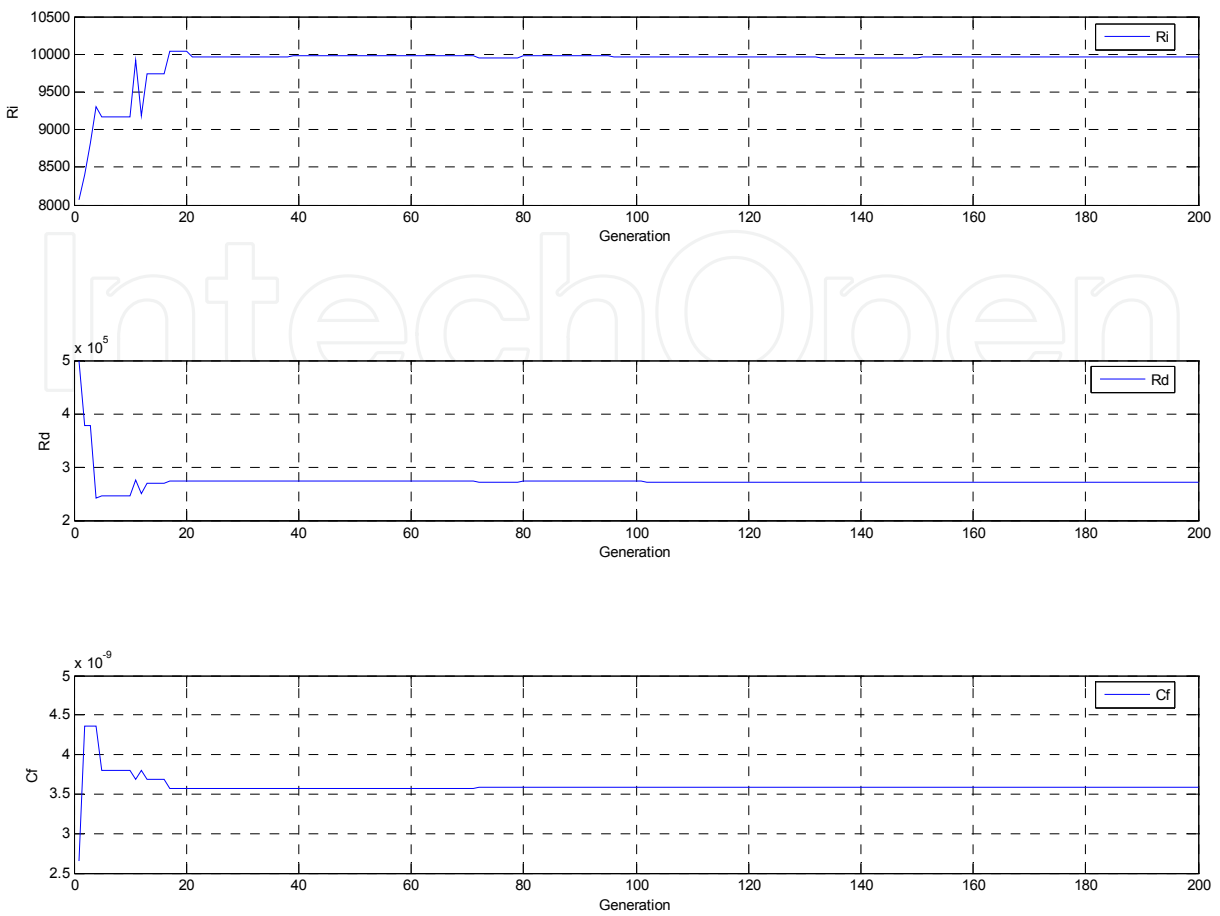


Figure 10. Trends of the optimum parameters of each generation

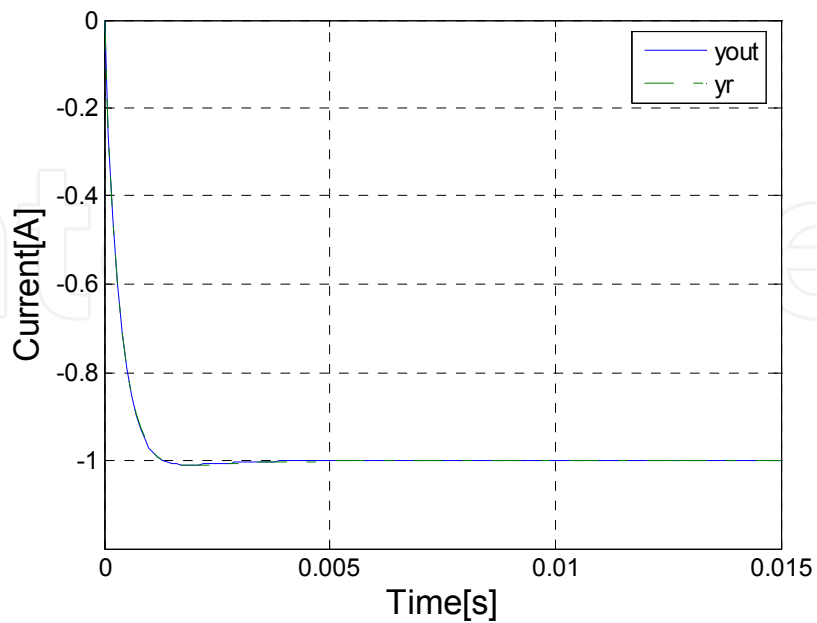


Figure 11. Step response comparison between the system obtained from the RCGA results and the system defined by the designer

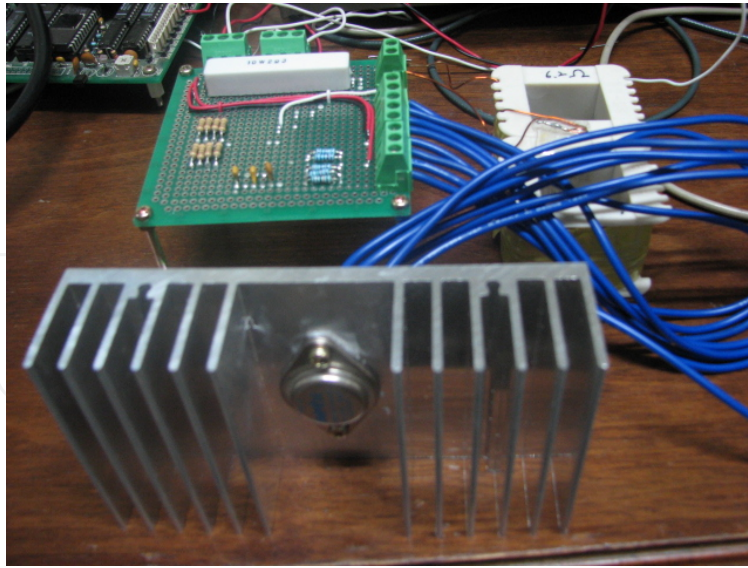


Figure 12. Amplifier applied with the designed parameter value

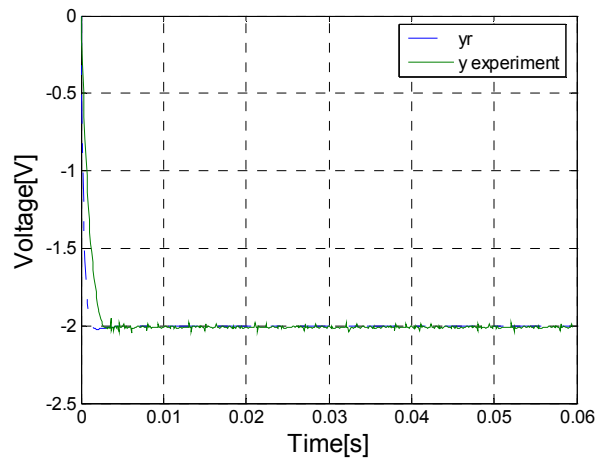


Figure 13. Step response comparison between the system that obtained from the RCGA results and the manufactured amplifier

Appendix 1 is a program that estimates the amplifier peripheral circuit part values in accordance with the response of the system defined arbitrarily, and the result is shown in Fig. 10, Fig. 9, Fig. 11 and Fig. 12 shows the step response measurement graphs of the system that was produced by designing and manufacturing the amplifier circuit using RCGA like that of Appendix 1.

3.3. Magnetic bearing system identification

In order to design the controller for the designed magnetic bearing system, there needs to be a process to estimate the parameter values that exist in the given system and is difficult to measure. In case a magnetic bearing system model with the same response as that of experimentation results can be found, the designer can design the desired controller without performing experimentation. Such a process is called identification.

In order to identify the magnetic bearing system, experimentation data of the manufactured system is necessary. However, since the magnetic bearing system is an unstable system, a controller to stabilize the system to obtain experimental data from the experiment device is necessary.

3.3.1. Implementation of the control signal

To control the magnetic bearing system, there is a need to stabilize the whole system by controlling the current flowing in the current. To do this, PID controller is introduced. In order to control the current of the coil using a PID controller, the voltage going into the current amplifier has to be controlled, and to implement such a voltage signal, a PID controller that is supported by Labview or Matlab has to be used or one that is discretized to match the sample time has to be used. Here, how a discretized PID controller is implemented is described.

Generally, a given PID controller can be defined as shown in Equation (27).

$$\frac{u(s)}{e(s)} = K_p + \frac{K_I}{s} + K_D s \quad (27)$$

Converting Equation (27) to z results in Equation (28).

$$\frac{u(z)}{e(z)} = K_p + \frac{K_I T(z+1)}{2(z-1)} + \frac{K_D(z-1)}{Tz} \quad (28)$$

Reducing Equation (28) and reorganizing $u(z)$ about gives Equation (29).

$$u(z) = \frac{2K_p Tz(z-1) + K_I T^2 z(z-1) + 2K_D(z-1)^2}{2Tz(z-1)}$$

$$(z^2 - z)u(z) = \left(\left(K_p + \frac{T}{2} K_I + \frac{K_D}{T} \right) z^2 + \left(\frac{T}{2} K_I - K_p - \frac{2K_D}{T} \right) z + \frac{K_D}{T} \right) e(z) \quad (29)$$

Taking into consideration that the z operator is a shift operator, Equation (29) is shown in cyclic form of Equation (30).

$$u(n+2) = \left(K_p + \frac{T}{2} K_I + \frac{K_D}{T} \right) e(n+2) + \left(\frac{T}{2} K_I - K_p - \frac{2K_D}{T} \right) e(n+1) + \frac{K_D}{T} e(n) + u(n+1) \quad (30)$$

Here, $e(n)$ is the nth sample data of the error signal.

If $K_p = 1$, $K_D = 0.005$, $K_I = 2$ and the sampling time T is assumed to be 0.001s, Equation (30) can be represented as Equation (31).

$$u(n+2) = 6.001e(n+2) - 10.999e(n+1) + 5e(n) + u(n+1) \quad (31)$$

If the PID controller can be shown in cyclic form, this can be conveniently programmed using C. Even here, the sampling time of the overall program has to be programmed equal to the sampling time of the PID controller.

3.3.2. Parameter identification

With regard to the whole system including the PID controller, the transfer function from the reference input r to the displacement x of the levitating object is as shown in Equation (32).

$$G_T = \frac{b_{m3}s^3 + b_{m2}s^2 + b_{m1}s + b_{m0}}{a_{m5}s^5 + a_{m4}s^4 + a_{m3}s^3 + a_{m2}s^2 + a_{m1}s + a_{m0}} \quad (32)$$

Here,

$$a_{m5} = R_I R_F C_f L m$$

$$a_{m4} = R_I R_F C_f m (R + R_s) + R_I R_d C_f m$$

$$a_{m3} = R_I m - R_I R_F C_f (bcR_s + aL) - bR_s R_F R_d C_f K_D G_s$$

$$a_{m2} = -a(R_I R_F C_f (R + R_s) + R_I R_d C_f) - bR_s R_F (K_D + R_d C_f K_P) G_s$$

$$a_{m1} = -aR_I - bR_s R_F (K_P + R_d C_f K_I) G_s$$

$$a_{m0} = -bR_s R_F K_I G_s$$

$$b_{m3} = -bR_s R_F R_d C_f K_D$$

$$b_{m2} = -bR_s R_F (K_D + R_d C_f K_P)$$

$$b_{m1} = -bR_s R_F (K_P + R_d C_f K_I)$$

$$b_{m0} = -bR_s R_F K_I$$

$$a = \frac{2kI_{ss}^2}{X_1^3}$$

$$b = -\frac{2kI_{ss}}{X_1^2}$$

$$c = \frac{2kI_{ss}}{X_1^2}$$

$$X_1 = X_0 + W + x$$

$$X_0 = \frac{l_m}{2\mu_s}$$

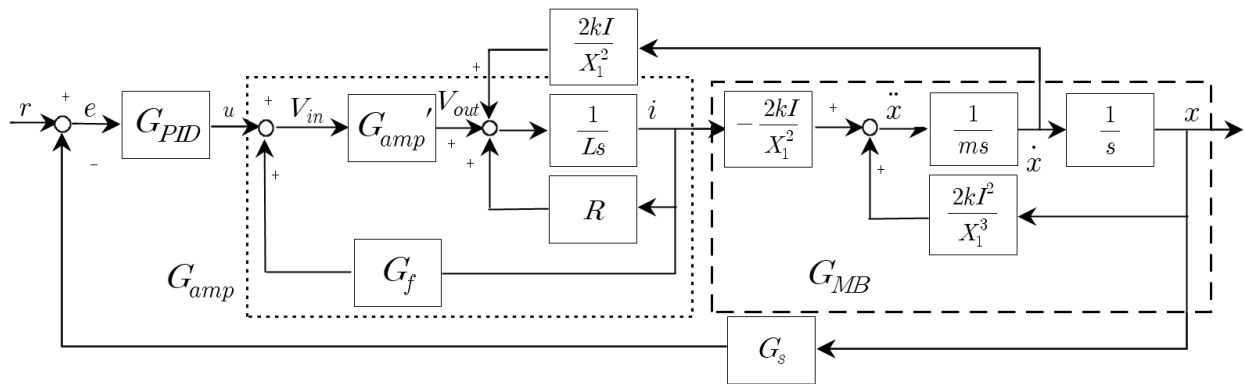


Figure 14. Block diagram of the magnetic bearing system including PID controller

Fig. 14 shows the block diagram of the whole system including the PID controller. To identify the unknown parameters using the genetic algorithm, the parameters difficult to measure has to be defined from the system transfer function and the search range of each parameter has to be defined. Using the error between the step response of the whole system and the data value obtained from experimentation, the objective function of the genetic algorithm is defined and the parameters of the genetic algorithm are defined.

Appendix 2 is the genetic algorithm program that allows the calculation of the unknown parameters from the above processes. At this point, the program part that is the same with the program to solve the amplifier peripheral circuit was excluded. Fig. 15 and Fig. 16 shows the graphs of the output results of the magnetic bearing system identification using RCGA similarly with Appendix 2.

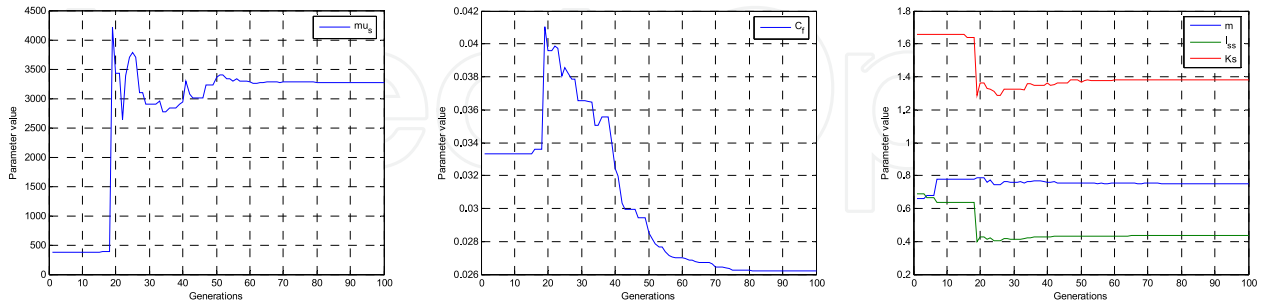


Figure 15. Trends of optimum parameters for each generation

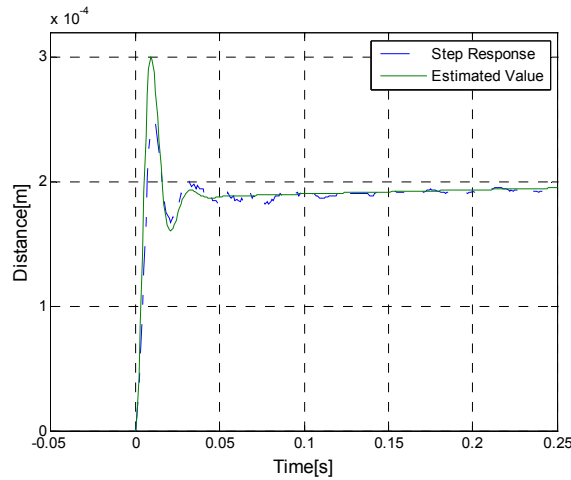


Figure 16. Experiment data comparison with the system obtained through RCGA results

3.4. Control signal division

The system of Fig. 14 is a modeling of the case where the magnetic bearing system is assumed to a horizontally symmetric based on the center point so that the levitating object is supported using an upper based electromagnet about the left or right parts. In order to properly levitate the levitating object of this system, a control signal equal to that of Fig. 14 has to be implemented and consistently supplied to the left and right magnetic bearing.

Also, to divide the control signal that supports the levitating object using only the upper electromagnet like that of Fig. 14 into the upper and lower electromagnet, the current i_{up} flowing in the upper electromagnet coil has to include the attractive force caused by the current i_{down} flowing in the lower electromagnet coil, where Equation (33) has to be followed for the design.

$$i_{up} = I_{ss} + i_{up0} + \alpha \tag{33}$$

Here, i_{up0} is the control current necessary to support the levitating object with only the upper electromagnet and α is the current corresponding to the attractive force caused by the current flowing in the lower electromagnet coil.

At this point, the magnetic bearing system has a form symmetric about each axis. Therefore, when assuming the axis parallel to the direction vertical from the Earth as the x-axis, the attractive force control of the y- and z-axes is the same as the x-axis control case including the neutral state and excluding the control current I_{ss} .

Fig. 17 is step response test result that is an example. When the program is work, levitated object is attracted by upper electric magnet. After that, the control signal is separated between upper and lower electric magnet. In the step response test, the disturbance mass is 150g.

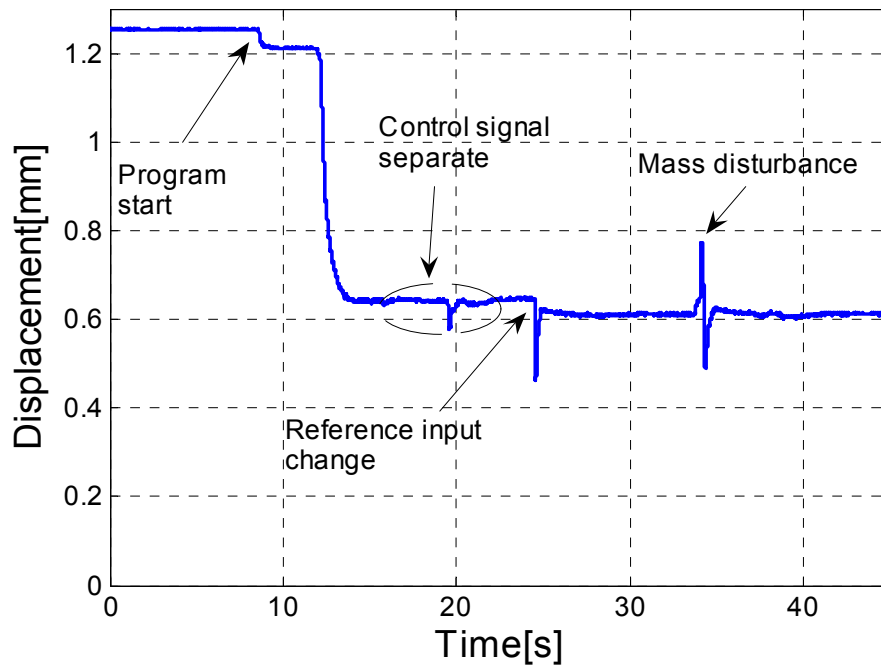


Figure 17. Step response test result for levitating system

3.5. Levitating object and equation of rotational motion

Fig. 18 shows a schematic of a magnetic bearing system taking into consideration rotational motion.

When the levitating object undergoes rotational motion, the torque caused by the rotational motion about the x-z plane is as shown in Equation (34).

$$J\ddot{\theta}_y - J_p\omega\dot{\theta}_x = -lf_{xl} + lf_{xr} \quad (34)$$

Here, J is the moment of inertia of the levitating object about the y-axis and J_p is the moment of inertia of the rotating levitating object about the x-axis. From Fig. 16, Equations (35) and (36) can be obtained.

$$\sin\theta_x = \frac{\Delta x}{l} \approx \theta_x \tag{35}$$

$$\sin\theta_y = \frac{\Delta y}{l} \approx \theta_y \tag{36}$$

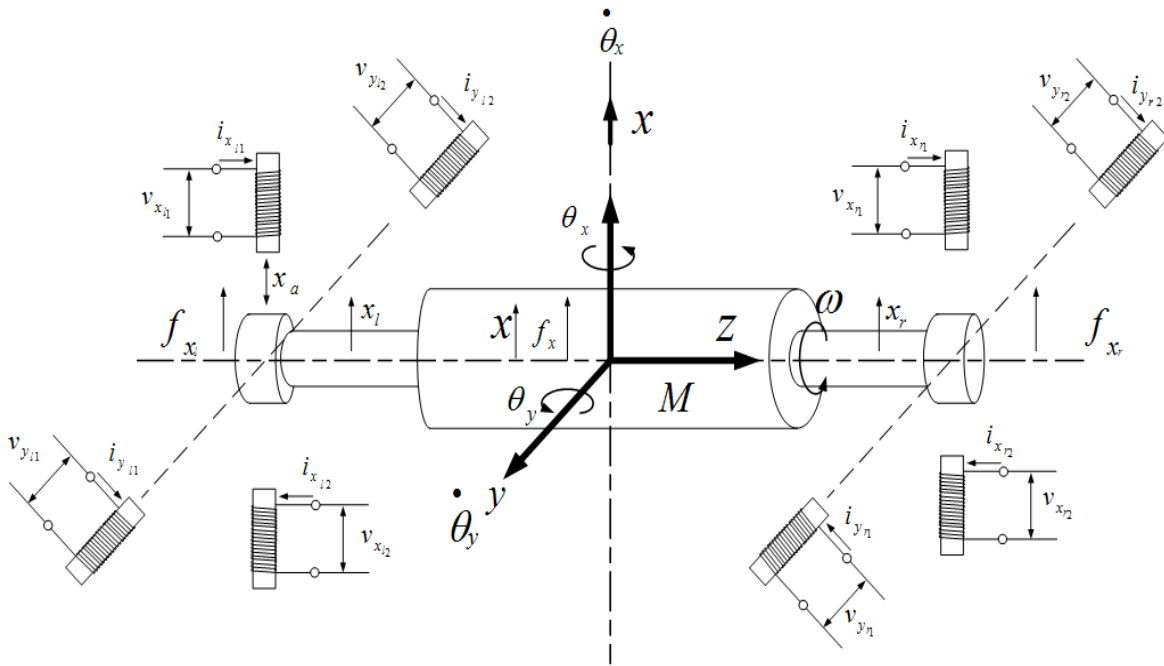


Figure 18. Magnetic bearing system taking into consideration rotational motion

When the levitating object undergoes rotational motion, the torque caused by the rotational motion about the y - z plane is as shown in Equation (37).

$$J\ddot{\theta}_x - J_p\omega\dot{\theta}_y = -lf_{y1} + lf_{y2} \tag{37}$$

In order to control the rotating levitating object, application of a multi-variable controller using state-space expression is necessary.

4. Conclusion

In this chapter, the detailed control laws to control the magnetic bearing was not included, and the method to designing a magnetic bearing system, obtaining a mathematical model, and the preparations necessary for control were explored with the aim of implementing a magnetic bearing system to apply various controllers.

Appendix

1. Attractive Force Calculation of the Electromagnet Using Probable Flux Paths Method
2. RCGA Program for Amplifier Peripheral Circuit Design
3. RCGA Program for Magnetic Bearing System Identification

1. Attractive force calculation of the electromagnet using probable flux paths method

It is assumed that the levitating object is supported by the electromagnet. Here, the gap between the electromagnet and the levitating object is 0.6mm, the number of coil winding to the electromagnet is 400turns, and the current flowing in the coil is 1A.

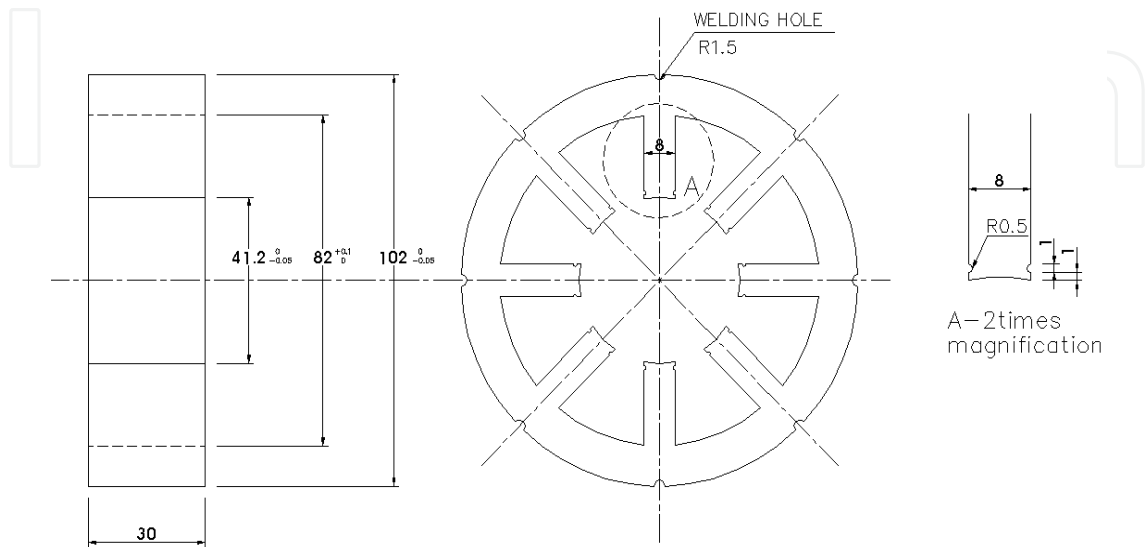


Figure 19. Electromagnet core drawing

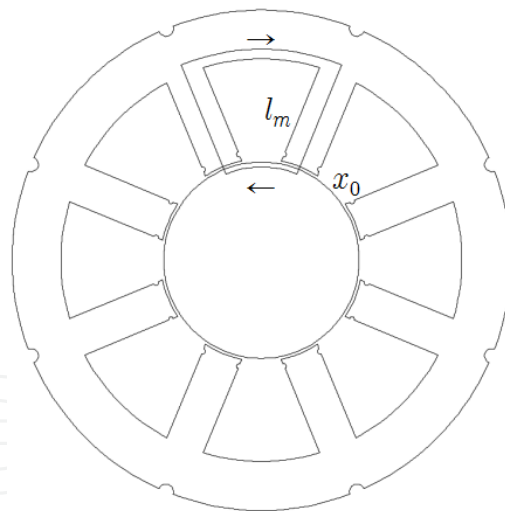


Figure 20. Magnetic circuit of the electromagnet

Fig. 19 shows the drawing of the electromagnet core. Fig. 20 shows the magnetic circuit that satisfies the Probable Flux Paths Method in the core of Fig. 19. The attractive force of the electromagnet is as shown in Equation (38).

$$F_m = \frac{\mu_0 N^2 I_m^2 S}{\left(\frac{l_m}{\mu_s} + 2x_0\right)^2} \quad (38)$$

Here, the length of the magnetic path is as shown in Equation (39) according to Fig. 2.

$$l_m = \frac{\pi d_1}{8} + \frac{\pi d_2}{8} + 2h = \left(\frac{\pi(82+41.2)}{8} + 2 \times 40 \right) \times 10^{-3} [\text{m}] \tag{39}$$

When assuming the material of the electromagnet core as silicon steel plate ($\mu_s = 3000$), the attractive force of this electromagnet is calculated using Equation (40).

$$F_m = \frac{4\pi \times 10^{-7} \times 400^2 \times 1^2 \times 2 \times 30 \times 10^{-3} \times 0.8 \times 10^{-3}}{\left(\left(\frac{\pi(82+41.2)}{8 \times 3000} + 2 \times 40 + 2 \times 0.6 \right) \times 10^{-3} \right)^2} = 6.2485 [\text{N}] \tag{40}$$

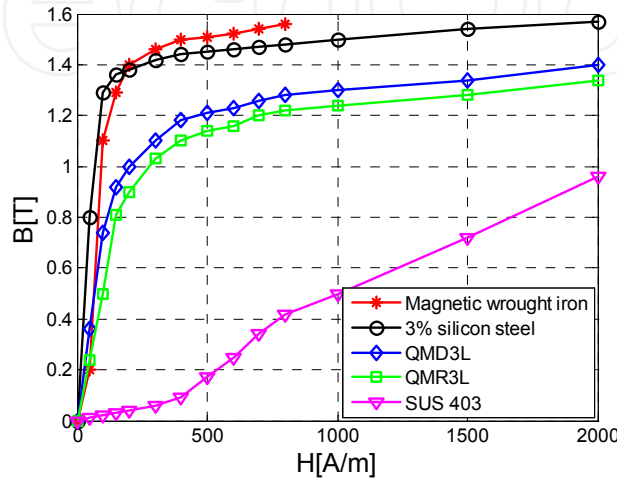


Figure 21. B-H curve of a magnetic substance

Fig. 21 shows the B-H curve of generic magnetic substances. As can be observed in Fig. 21, the magnetic flux density is concentrated about the magnetomotive force above a certain level for magnetic substances, implying that the attractive force does not increase when the current of the coil is increased to increase the magnetomotive force as the magnetic flux density does not increase. Therefore, in order to identify the maximum attractive force of an electromagnet, a process in identifying the maximum attractive force that can be used from the saturated magnetic flux density of the electromagnet material or the maximum current that can be bled in the coil is necessary.

The relationship between the current flowing in the coil and the magnetic flux density is as shown in Equation (41).

$$B = \frac{\Phi}{S} = \frac{\mu_0 N I_m}{\frac{l_m}{\mu_s} + 2x_0} \tag{41}$$

From this, the maximum current I_{max} from the saturated magnetic flux density B_{max} can be solved as shown in Equation (42). However, the saturated magnetic flux density of silicon steel is approximately 1.5T.

$$I_{max} = \frac{\left(\frac{l_m}{\mu_s} + 2x_0 \right) B_{max}}{\mu_0 N}$$

$$I_{max} = \frac{\left(\frac{\pi(82+41.2)}{8 \times 3000} + 2 \times 40 + 2 \times 0.6 \right) \times 10^{-3} \times 1.5}{4\pi \times 10^{-7} \times 400} = 3.7087 [\text{A}] \tag{42}$$

Here, the electromagnet maximum attractive force when the gap is 0.6mm is as shown in Equation (43).

$$F_{\max} = \frac{4\pi \times 10^{-7} \times 400^2 \times 3.7087^2 \times 2 \times 30 \times 10^{-3} \times 0.8 \times 10^{-3}}{\left(\left(\frac{\pi(82+41.2)}{8 \times 3000} + 2 \times 40 + 2 \times 0.6\right) \times 10^{-3}\right)^2} = 85.9442[\text{N}] \quad (43)$$

2. RCGA program for amplifier peripheral circuit design

Fig. 22 shows the linear current amplifier circuit. Considering that the voltage in Fig. 22 is V'_{fb} , the current can be expressed as Equation (44) and Equation (45) can be rearranged to obtain Equation (45).

$$i_1 = i_2 + i_3 \quad (44)$$

$$\frac{V_{IN} - V_{fb}}{R_I} = \frac{V_{fb} - V'_{fb}}{R_F} + \frac{V_{fb} - V_o}{Z_F} \quad (45)$$

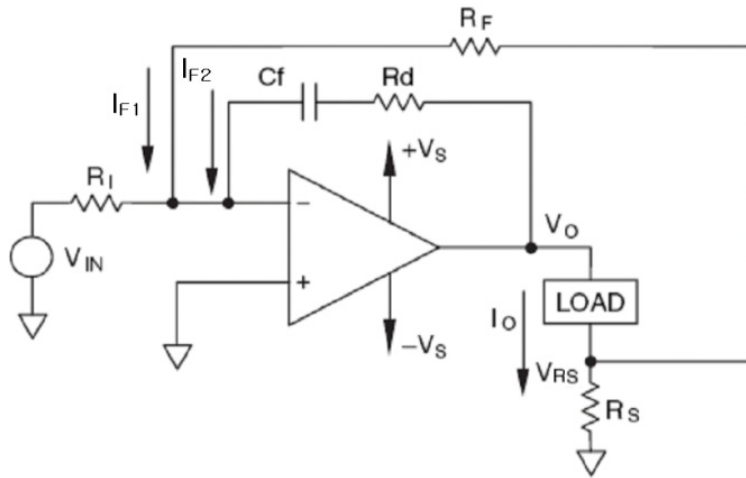


Figure 22. Linear current amplifier circuit

Also, considering that the voltage in Fig. 22 is V'_{fb} , the current can be expressed as Equation (46), and Equation (46) can be rearranged to obtain Equation (47).

$$i_2 + I_o = i_4 \quad (46)$$

$$\frac{V_{IN} - V'_{fb}}{R_F} + I_o = \frac{V'_{fb} - 0}{R_S} \quad (47)$$

Equation (48) is obtained from the open-loop gain of the operational amplifier and the relationship of Equation (49) is obtained from the current flowing in the load.

$$V_{fb} = -\frac{V_o}{A} \quad (48)$$

$$V'_{fb} = V_o - I_o Z_L \quad (49)$$

When the transfer function is found from the relationships of Equations (45), (47), and (48), it is as shown in Equation (50).

$$\frac{I_o}{V_{IN}} = \frac{n'}{d'} \tag{50}$$

Here,

$$\begin{aligned} n' &= -n_1s - n_0 \\ n_1 &= \left(\frac{R_s}{A} + R_s + R_F\right) R_F R_d C_f \\ n_0 &= \left(\frac{R_s}{A} + R_s + R_F\right) R_F \\ d' &= (a_1 a_2 - a_3) s^2 + (a_1 b_1 + a_2 b_2 - b_3) s + b_1 b_2 - c_1 \\ a_1 &= b_2 R_d C_f + \frac{R_I R_F}{A} C_f + R_I R_F C_f \\ a_2 &= (R_s + R_F) L \\ a_3 &= \left(\frac{R_s}{A} + R_s + R_F\right) R_I R_d C_f L \\ b_1 &= R_s R + R_F R_s + R_F R \\ b_2 &= \frac{R_F}{A} + \frac{R_I}{A} + R_I \\ b_3 &= \left(\frac{R_s}{A} + R_s + R_F\right) R_I (R_d C_f R + L) \\ c_1 &= \left(\frac{R_s}{A} + R_s + R_F\right) R_I R \end{aligned}$$

When A is assumed to be sufficiently large, Equations (50) and (23) are equivalent.

Before selecting the current amplifier circuit part values, the parts to find the value of are first identified. The parts to determine first are the amplifier (-)input resistance R_I and the feedback line resistance R_F . These resistances are parameters that determine the amplification ratio of the amplifier output current and are resistances to transfer the voltage signal. R_I and R_F has the relationship of Equation (51) from the DC gain of the closed circuit transfer function.

$$R_F = -R_I R_s \frac{I_o}{V_{IN}} \tag{51}$$

The ratio of the current amplifier input voltage and output current to be designed is assumed to be 1 : 1 and accordingly, R_F has the same relationship as shown in Equation (52).

$$R_F = -R_I R_s \tag{52}$$

The current amplifier amplification ratio A, current limiting resistance R_s , load inductance L and resistance R values are given as organized in Table. 1. Therefore, the variables of genetic algorithm to determine are limited to the resistance R_I which determines the amplifier output current amplification ratio, resistance R_d which determines the dynamic characteristics of the linear amplifier circuit, and condenser C_f .

Parameter	Value
A	$10^{107/20}$
L	9.2mH
R	6.2Ω
R_s	2Ω

Table 1. The known parameter's value at the current amplifier

In the case of resistance R_l which determines the amplifier output current amplification ratio, it has to be sought after in the range of $k\Omega$. In comparison, resistance R_d , which determines the dynamic characteristics of the linear amplifier circuit, has to be sought after in a wide range. It is desirable to use a value for the condenser C_f which determines the dynamic characteristics of linear amplifier circuit in the nF to μF range in consideration of the dynamic characteristics of the current amplifier.

Therefore, the search ranges are determined as shown in Equations (53), (54), and (55).

$$8000 \leq R_l \leq 11000 \quad (53)$$

$$10^{-1} \leq R_d \leq 10^6 \quad (54)$$

$$10^{-10} \leq C_f \leq 10^{-7} \quad (55)$$

The transfer function with a desired output is defined as shown in Equation (56) with the same form as the transfer function of a current amplifier circuit. This is so that the desired output can be achieved with the combination of part values of the current amplifier circuit.

$$G_r(s) = -\frac{e_1 s + e_0}{s^2 + d_1 s + d_0} \quad (56)$$

Equation (57) was obtained through trial and error in an effort to obtain a step response rise time below 0.001s and the percentage overshoot below 5%. Fig. 23 shows the step response of the transfer function of Equation (57).

$$G_r(s) = -\frac{3000s + 210000}{s^2 + 3600s + 210000} \quad (57)$$

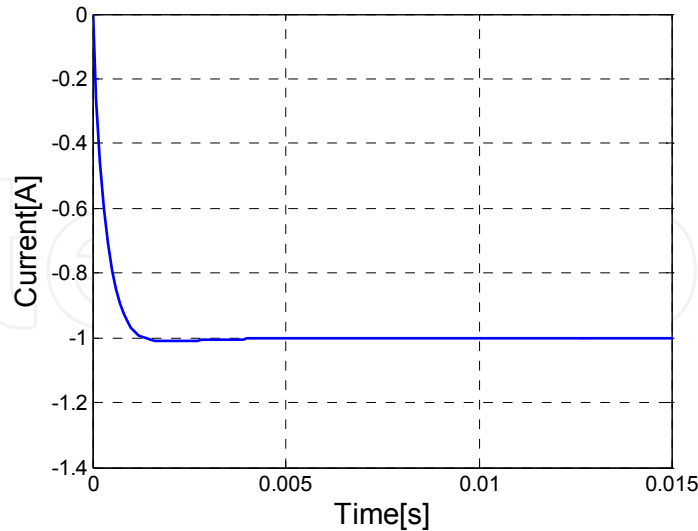


Figure 23. The desired step response

The objective function to apply genetic algorithm is Equation (58).

$$F_{obj} = \int e(t)^2 dt \quad (58)$$

Here, $e(t) = y_r(t) - y_{out}(t)$, $y_r(t)$ is the step response of the system satisfying the required time response characteristics and $y_{out}(t)$ is the current amplifier circuit transfer function step response obtained from the selected chromosome.

Table 2 shows the parameter values necessary to implement the real coded genetic algorithm.

The real coded genetic algorithm problem was implemented through matlab and the solution could be found by executing the rcga.m file. The names of files linked to rcga.m and their functions are shown in Table 3. Each script is as follows.

Parameter	Parameter value	Genetic operator
Population	50	Generate initial population
Max. generation	200	Generate initial population
Chromosome length	3	Generate initial population
Crossover probability	0.9	Modified simple crossover
Mutation probability	0.1	Dynamic mutation
eta	1.7	Scale fitting

Table 2. The parameter's value of the RCGA for the current amplifier design

file name (*.m)	function
rcga	Find optimal value on object function with RCGA
rInitPa	Define program variable for RCGA
rInitPop	Initialize the population
EvalObj	Evaluate the object function on the population for the reproduction
rGradSel	Reproduction operator with a gradient like selection method
rMsXover	Crossover operator with a modified simple crossover method
rDynaMut	Mutation operator with a dynamic mutation method
rElitism	Let to survive the best chromosome at the present generation to next generation
ScaleFit	To improve the reproduction operator's efficiency
rStatPop	Memorize the population's state for each generation

Table 3. Program file list for the execution of the real coded genetic algorithm

```

% rcga.m
% The RCGA implements a real coded genetic algorithm for finding the
% component value in the current amp. circuit
%
% Encoding:
%   - Real
%
% Genetic operators:
%   - Gradient-like selection
%   - Modified simple crossover
%   - Dynamic mutation
%
% Other strategies:
%   - Elitism
%   - scaling window scheme(Ws=1)
%
% Remarks:
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNNU

clear;

% initializes the generation counter
gen= 1;

% initializes the parameters of a RCGA
[rseed,maxmin,maxgen,popsize,lchrom,pcross,pmutat,xlb,xub,etha,Ev]=
rInitPa;

% creates a polulation randomly
pop= rInitPop(rseed,popsize,lchrom,xlb,xub);

% calculates the objective function value
objfunc= EvalObj_new3(pop,lchrom,popsize);

% calculates gam
if(maxmin == 1)
    gam= min(objfunc);
else
    gam= min(-objfunc);
end

% calculates fitness using the scaling window scheme
fitness= ScaleFit(objfunc,popsize,gam,maxmin);

% computes statistics
[chrombest,objbest,fitbest,objave,gam]=
rStatPop(pop,objfunc,fitness,maxmin);

```

```

% builds a matrix storage for plotting line graphs
stats(gen,:)= [gen objbest objave chrombest];

for gen= 2:maxgen

% prints the current generation
    fprintf('gen= %d (%d)\n',gen,maxgen-gen);

% applies reproduction
    pop=
rGradSel(pop,popsize,lchrom,fitness,chrombest,fitbest,xlb,xub,etha);
% Gradient-like selection

% applies crossover
    [pop,nxover]= rMsXover(pop,popsize,lchrom,pcross); % modified
simple crossover

% applies mutation
    [pop,nmutat]=
rDynaMut(pop,popsize,lchrom,pmutat,xlb,xub,gen,maxgen); %dynamic
mutation

% calculates the objective function value
    objfunc= EvalObj_new3(pop,lchrom,popsize);

% applies modified Elitism
[pop,objfunc]= rElitism(pop,objfunc,chrombest,objbest,maxmin);

% applies the scaling window scheme
    fitness= ScaleFit(objfunc,popsize,gam,maxmin);

% computes statistics
    [chrombest,objbest,fitbest,objave,gam]=
rStatPop(pop,objfunc,fitness,maxmin);

% builds a matrix storage for plotting line graphs
    stats(gen,:)= [gen objbest objave chrombest];
end

figure(1)
% plots the best and average objective function values
subplot(2,1,1)
plot(stats(:,1),stats(:,2))
xlabel('Generation'),ylabel('object function')

% plots the variables of the best chromosome
subplot(2,1,2)
plot(stats(:,1),stats(:,4),'-',stats(:,1),stats(:,5),'--
',stats(:,1),stats(:,6),'--')
xlabel('Generation'),ylabel('control parameter')

```



```

legend('R1','Rd','Cf')

figure(2)

Rs = 2; R = 6.2; A = 10^(107/20); L = 9.2 * 10^-3;

i=1;
R1 = chrombest(1,1) ;
Rf = Rs * R1;
var(i,1) = chrombest(1,2);
var(i,2) = chrombest(1,3);

a = -(Rs/A +Rs +Rf)*Rf*var(i,1)*var(i,2);
b = -(Rs/A +Rs +Rf)*Rf;
c = ((Rf/A + R1/A +R1)*var(i,1)*var(i,2) + R1*Rf*var(i,2)/A +
R1*Rf*var(i,2))*(Rs + Rf)*L - (Rs/A + Rs
+Rf)*R1*var(i,1)*var(i,2)*L;
d1 = ((Rf/A + R1/A +R1)*var(i,1)*var(i,2) + R1*Rf*var(i,2)/A +
R1*Rf*var(i,2))*(Rs*R + Rf*Rs + Rf*R) + (Rs + Rf)*L*(Rf/A + R1/A
+R1) - (Rs/A + Rs +Rf)*R1*(var(i,1)*var(i,2)*R + L);
e = (Rf/A + R1/A +R1)*(Rs*R + Rf*Rs + Rf*R) - (Rs/A + Rs +Rf)*R1*R;

n=[a b];
d=[c d1 e];
h= 0.0001; wdata = 150; t=0:h:wdata*h;

yn=[3000 2100000];
yd=[1 3600 2100000];
r2=step(-yn,yd,t);
r1=step(n,d,t);

plot(t,r1,'-',t,r2,'--')
legend('yout','yr')
xlabel('Time[s]'),ylabel('Current[A]')

% rInitPa.m
% The RINITPA function initializes the parameters of a RCGA
%
% Output:
%   rseed- random seed
%   maxmin= -1 for minimization, 1 for maximization
%   maxgen-   maximum generation
%   popsize- population size(must be an even integer)
%   lchrom- chromosome length
%   pcross- crossover probability
%   pmutat- mutation probability
%   xlb- lower bound  of variables
%   xub- upper bound of variables
%   etha- parameter of the selection operator

```

```

%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKN

function
[rseed,maxmin,maxgen,popsize,lchrom,pcross,pmutat,xlb,xub,etha,Ev]=
rInitPa

rseed=      8512;
maxmin=     -1;           % -1 for minimization
maxgen=     200;
popsize=    100;         % popsize should be even
lchrom=      3;
etha=       1.7;
pcross=     0.9;
pmutat=     0.1;
xlb(1,1) = 8000;
xlb(1,2) = 0.1;
xlb(1,3) = 1*10^-10;

xub(1,1) = 11000;
xub(1,2) = 1000000;
xub(1,3) = 1*10^-7;

Ev=0;

if(rem(popsize, 2) ~= 0) % do not move
    popsize= popsize + 1;
end

% rInitPop.m
%
% The RINITPOP function creates an initial population
%
% Input:
%   rseed- random seed
%   popsize- population size
%   lchrom- chromosome length
%   xub- upper bound for variables, vector
%   xlb- lower bound for variables, vector
% Output:
%   pop- population
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKN

```

```

function pop= rInitPop(rseed,popsize,lchrom,xlb,xub)

rand('seed',rseed);
pop= zeros(popsize,lchrom);
for i=1:popsize
    pop(i,:)= (xub-xlb).*rand(1,lchrom)+xlb;
end

% EvalObj_new3.m
%
% The EVALOBJ function evaluates the objective function value
%
% Input:
%   var- variables, matrix
%   npara- number of the variables
%   popsize- population size
% Output:
%   objfunc- objective function value, vector
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function objfunc= EvalObj_new3(var,npara,popsize);

Rs = 2; R = 6.2; A = 10^(107/20); L = 9.2 * 10^-3;

for i= 1:popsize
objfunc(i)=0; oldobj=0;
R1 = var(i,1);
Rf = Rs * R1;
a = -(Rs/A +Rs +Rf)*Rf*var(i,2)*var(i,3);
b = -(Rs/A +Rs +Rf)*Rf;
c = ((Rf/A + R1/A +R1)*var(i,2)*var(i,3) + R1*Rf*var(i,3)/A +
R1*Rf*var(i,3))*(Rs + Rf)*L - (Rs/A + Rs
+Rf)*R1*var(i,2)*var(i,3)*L;
d1 = ((Rf/A + R1/A +R1)*var(i,2)*var(i,3) + R1*Rf*var(i,3)/A +
R1*Rf*var(i,3))*(Rs*R + Rf*Rs + Rf*R) + (Rs + Rf)*L*(Rf/A + R1/A
+R1) - (Rs/A + Rs +Rf)*R1*(var(i,2)*var(i,3)*R + L);
e = (Rf/A + R1/A +R1)*(Rs*R + Rf*Rs + Rf*R) - (Rs/A + Rs +Rf)*R1*R;

n=[a b];
d=[c d1 e];

h= 0.0001; wdata = 150; t=0:h:wdata*h;
yn=[-3000 -2100000];
yd=[1 3600 2100000];
yr = step(yn,yd,t);
resp = step(n,d,t);
err(:,1) = resp(:,1) - yr(:,1);

```

```

for j= 1:wdata
    obj= err(j,1)^2;
    objfunc(i)= objfunc(i)+0.5*h*(obj+oldobj);
    oldobj= obj;
end
end

% ScaleFit.m
%
% The SCALEFIT function converts objective function values into
fitness using
% the scaling window scheme(window size= 1)
%
% Input:
%   objfunc- objective function value, vector
%   popsize- population size
%   gam- minimum of objfunc or -objfunc in the previous population
%   maxmin= -1 for minimization, 1 for maximization
% Output:
%   fitness- scaled fitness, vector
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function fitness= ScaleFit(objfunc,popsize,gam,maxmin)

if(maxmin == 1)
    fitness= objfunc-gam;
else
    fitness= -objfunc-gam;
end
for i=1:popsize
    if(fitness(i) < 0)
        fitness(i)= 0;
    end
end

% rStatPop.m
%
% The RSTATPOP function calculates the statistics of a population
%
% Input:
%   pop- population, matrix
%   objfunc- objective function value, vector
%   fitness- fitness, vector
%   maxmin= -1 for minimization, 1 for maximization

```

```

% Output:
% chrombest- best chromosome, vector
% objbest- best objective function value
% fitbest- fitness of the best chromosome
% objave- average objective function value
% gam- minimum of objfunc or -objfunc
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function [chrombest,objbest,fitbest,objave,gam]=
rStatPop(pop,objfunc, ...

fitness,maxmin)

if(maxmin == 1)
    [objbest, index]= max(objfunc);
    gam= min(objfunc);
else
    [objbest, index]= min(objfunc);
    gam= min(-objfunc);
end
chrombest= pop(index,:);
fitbest= fitness(index);
objave= mean(objfunc);

% rGradSel.m
%
% The RGRADSEL function performs gradient-like selection
%
% Input:
% pop- population of chromosomes, matrix
% popsize- population size
% lchrom- chromosome length
% fitness- fitness, vector
% chrombest- best chromosome, vector
% fitbest- fitness of the best chromosome
% xlb- lower bound for variables, vector
% xub- upper bound for variables, vector
% etha- parameter of the selection operator
% Output:
% newpop- mating pool, matrix
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

```

```

function newpop=
rGradSel(pop,popsize,lchrom,fitness,chrombest,fitbest,xlb, ...

xub,etha)
if(fitbest > 0)
    for i= 1:popsize
        etha1= etha;
        normfit= 1-fitness(i)/fitbest;
        pass= 0;
        while(pass == 0)
            pass= 1;
            for j= 1:lchrom
                newpop(i,j)= pop(i,j)+etha1*normfit*(chrombest(j)-
pop(i,j));
                if(newpop(i,j) < xlb(j) | newpop(i,j) > xub(j))
                    etha1= etha1*0.8;
                    pass= 0;
                    break;
                end
            end
        end
    end
end
else
    for i= 1:popsize
        k= Pickup(popsize);
        newpop(i,:)= pop(k,:);
    end
end

% Pickup.m
%
% The PICKUP function picks up an integer random number between 1
and num
%
% Input:
%   num- integer number greater than or equal to 1
% Output:
%   rnum- random number between 1 and num
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function rnum= Pickup(num)

if min(num) < 1
    disp('num is less than one !')
    return;
end

```

```

fr= rand(size(num));
rnum= floor(fr.*num)+1;

% rMsXover.m
%
% The RMSXOVER function performs modified simple crossover
%
% Input:
%   pop- population of chromosomes, matrix
%   popsize- population size
%   lchrom- chromosome length
%   pcross- crossover probability
% Output:
%   pop- mated population, matrix
%   nxover- number of times crossover was performed
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function [pop,nxover]= rMsXover(pop,popsize,lchrom,pcross)

nxover= 0;
halfpop= floor(popsize/2);
for i= 1:halfpop
    if (rand <= pcross)
        nxover= nxover+1;
        mate1= 2*i-1;
        mate2= 2*i;
        xpoint= Pickup(lchrom-1);
        lam= rand;
        temp= lam*pop(mate2,xpoint)+(1-lam)*pop(mate1,xpoint);
        lam= rand;
        pop(mate2,xpoint)= lam*pop(mate1,xpoint)+(1-
lam)*pop(mate2,xpoint);
        pop(mate1,xpoint)= temp;

        temp= pop(mate1,xpoint+1:lchrom);
        pop(mate1,xpoint+1:lchrom)= pop(mate2,xpoint+1:lchrom);
        pop(mate2,xpoint+1:lchrom)= temp;
    end
end

% rDynaMut.m
%
% The RDYNAMUT function performs dynamic mutation
%
```

```

% Input:
%   pop- population of chromosomes, matrix
%   popsize- population size
%   lchrom- chromosome length
%   pmutat- mutation probability
%   xlb- lower bound of variables
%   xub- upper bound of variables
% Output:
%   pop- mutated population, matrix
%   nmutat- number of times mutation was performed
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

```

```

function [pop,nmutat]=
rDynaMut(pop,popsize,lchrom,pmutat,xlb,xub,gen,maxgen)

b= 5;
nmutat= 0;
for i= 1:popsize
    for j= 1:lchrom
        if (rand <= pmutat)
            nmutat= nmutat+1;
            r= rand;
            if(round(rand))
                pop(i,j)= pop(i,j)+(xub(j)-pop(i,j))*r*(1-
gen/maxgen)^b;
            else
                pop(i,j)= pop(i,j)-(pop(i,j)-xlb(j))*r*(1-
gen/maxgen)^b;
            end
        end
    end
end
end

```

```

% rElitism.m
%
% The RELITISM function performs elitism
%
% Input:
%   pop- population of chromosomes, matrix
%   objfunc- objective function value
%   chrombest- best chromosome, vector
%   objbest- best objective function value
%   maxmin= -1 for minimization, 1 for maximization
% Output:
%   pop- modified population of chromosomes, matrix
%   objfunc- modified objective function value

```



```

%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function [pop,objfunc]=
rElitism(pop,objfunc,chrombest,objbest,maxmin)

if(maxmin==1)
    cobjbest= max(objfunc);
    if(cobjbest < objbest)
        [objworst, index]= min(objfunc);
        pop(index,:)= chrombest;
        objfunc(index)= objbest;
    end
else
    cobjbest= min(objfunc);
    if(cobjbest > objbest)
        [objworst, index]= max(objfunc);
        pop(index,:)= chrombest;
        objfunc(index)= objbest;
    end
end
end

```

3. RCGA program for magnetic bearing system identification

The transfer function from the reference input of the magnetic bearing system including the PID controller to the displacement of the levitating object is as shown in Equation (59).

The PID controller coefficients selected for the stabilization of the magnetic bearing system are $K_p = 1$, $K_D = 0.005$, $K_I = 2$ and the sampling time is 0.001s. If the PID controller is expressed in cyclic form to implement as a micro processor, it is as shown in Equation (59).

$$u(n+2) = 6.001e(n+2) - 10.999e(n+1) + 5e(n) + u(n+1) \quad (59)$$

Fig. 25 is the step response that was obtained from the magnetic bearing system including the PID controller designed for stabilization. Specially, Fig. 25 is the step response of the displaced levitating object displacement x when the right side electromagnet reference input was modified from 0.4mm to 0.6mm where the left side electromagnet was fixed.

Fig. 60 shows the connection diagram of the magnetic bearing control system. The power of the system uses a DC power supply, displacement sensor amplifier, DSP, and AC220V power for the PC, and DC power is used for the current amplifier. The control system is connected to the magnetic bearing coil and displacement measurement sensor through a port. The delivered signal from the displacement sensor amplifier is compared to the reference input in the DSP and a control signal is generated, where the control signal generated in the DSP is provided to the linear current amplifier circuit to control the electromagnet. The signals occurring during control are stored in the independently installed PC through the DAQ board(PCI6010) for monitoring.

The MPU to implement the PID controller is TMS320C32, and a 12 bit A/D converter (MAX122) and 12 bit D/A converter (AD664) were used. Eddy current type sensor (AH-305) was used as the displacement measurement sensor for the feedback signal and an appropriate sensor amplifier (AS-440-01) was applied.

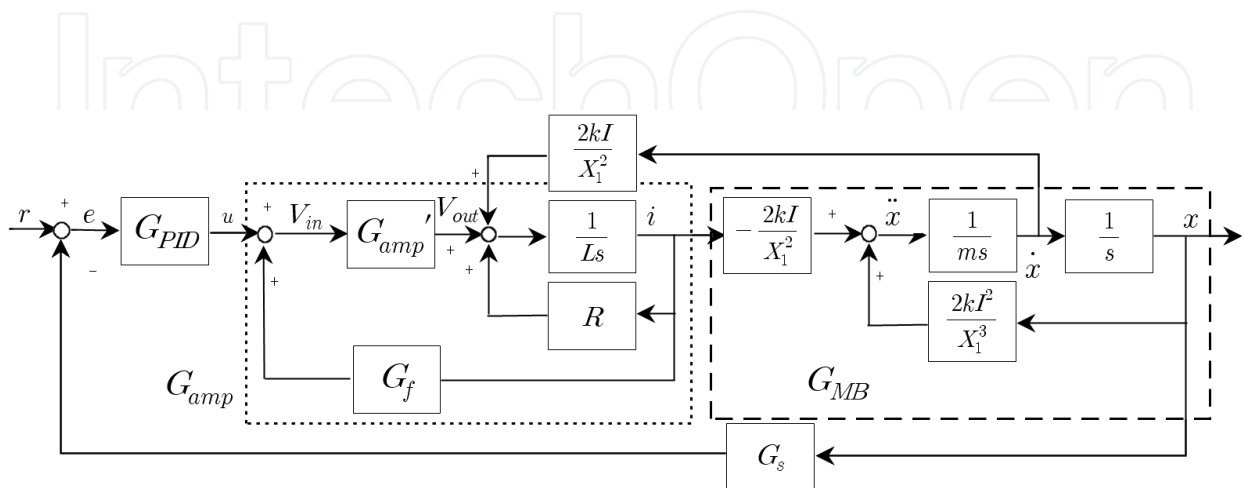


Figure 24. Magnetic bearing system including the PID controller

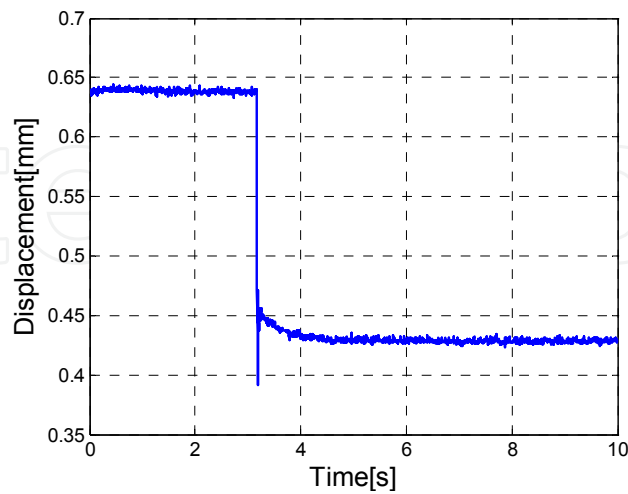


Figure 25. Step response of magnetic bearing system including the PID controller

Table 4 shows the parameter values already known regarding the magnetic bearing system. Therefore, the parameters that their values cannot be relatively exactly known in this system are the levitating object mass m , the current during normal state I_{ss} , coil inductance value L , relative permeability μ_s of the levitating object, and additional gain K_s for normal deviation calibration.

In the experiment for the identification of the magnetic bearing, the levitating object was supported using one side of the magnetic bearing. When supporting the levitating object with one side of the electromagnet, the levitating object becomes slanted so the vertical direction force that the electromagnet supports varies with the tilted angle of the levitating object and the impact force (mass m) of the levitating object on the electromagnet is difficult to measure. If the mass of the levitating object changes, the current I_{ss} at normal state depending on the mass also varies. Additionally, the coil inductance value L varies depending on the levitating object location within the electromagnet coil, thus, it is a parameter that is difficult to exactly measure. The relative permeability μ_s of the levitating object is also difficult to exactly obtain due to the uneven nature of the material, and the random gain to calibrate the normal deviation that occurs due to the mathematical model error is defined as K_s and is additionally included in the list of parameters to be identified.

Equation (60) shows the search ranges of the 5 unknown parameter values to be estimated by using the genetic algorithm. For each parameter, the search range was determined based on the actual experimented system and with the consideration of the physical characteristics.

$$\begin{aligned}
 0.5 &\leq m \leq 0.9 \\
 0 &\leq I_{ss} \leq 0.7 \\
 0 &\leq L \leq 0.1 \\
 0 &\leq \mu_s \leq 10000 \\
 0 &\leq K_s \leq 2
 \end{aligned} \tag{60}$$

Parameter	Value
Length of a path for magnetic flux	0.1711m
Displacement for levitate object(at steady state)	0.6mm
Cross section of armature	$4.8 \times 10^{-4} \text{ m}^2$
Number of coil turn for magnetic bearing	200 turn

Table 4. The table for the known parameters

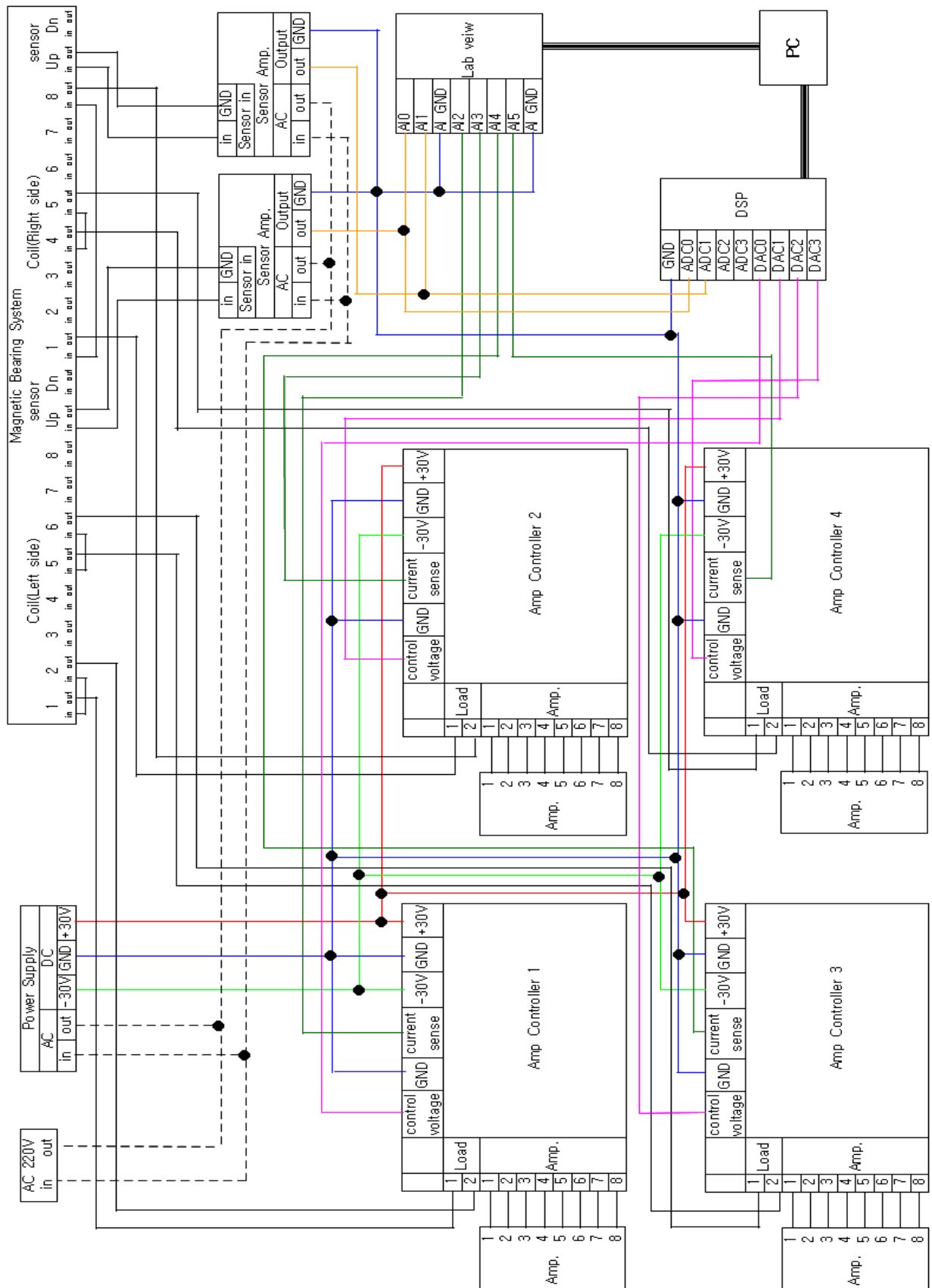


Figure 26. The connecting diagram for the magnetic bearing system

IAE(Integrated Absolute Error) as shown in Equation (61) was selected as the objective function to execute RCGA.

$$f_{obj} = \int_{t_0}^{t_f} |e(t)| dt \quad (61)$$

Here, $e(t)$ is the difference between the magnetic bearing step response obtained experimentally and the magnetic bearing model step response obtained mathematically through the selected chromosome.

For the implement the real coded genetic algorithm, the program and the modified parts of Table 3 are shown in Table 6 and the scripts are as follows.

Parameter	Parameter Value	Genetic operator
Population	100	Generate initial population
Max. generation	100	Generate initial population
Chromosome length	5	Generate initial population
Crossover Probability	0.9	Modified simple crossover
Mutation Probability	0.1	Dynamic mutation
Eta	1.7	Scale fitting

Table 5. Shows the program parameters necessary to execute RCGA.

file name (*.m)	function
rcga	Find optimal value on object function with RCGA
rInitPa	Define program variable for RCGA
EvalObj	evaluate the object function on the population for the reproduction

Table 6. Modified program file list to execute real coded genetic algorithm

```

% rcga22.m
%
% The RCGA22 implements a real coded genetic algorithm for finding
system parameter in the MBS
%
%   Encoding:
%   - Real
%
% Genetic operators:
%   - Gradient-like selection
%   - Modified simple crossover
%   - Dynamic mutation
%
% Other strategies:
%   - Elitism
%   - scaling window scheme (Ws=1)
%
% Remarks:
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

clf;
clear;
Test_data = xlsread('a_pidR.xls');

% initializes the generation counter
gen= 1;

% initializes the parameters of a RCGA
[rseed,maxmin,maxgen,popsize,lchrom,pcross,pmutat,xlb,xub,etha]=
rInitPa22;

% creates a polulation randomly
pop= rInitPop(rseed,popsize,lchrom,xlb,xub);

% calculates the objective function value
objfunc= EvalObj22(pop,lchrom,popsize,Test_data);

% calculates gam
if(maxmin == 1)
    gam= min(objfunc);
else
    gam= min(-objfunc);
end

% calculates fitness using the scaling window scheme
fitness= ScaleFit(objfunc,popsize,gam,maxmin);

```

```

% computes statistics
[chrombest,objbest,fitbest,objave,gam]=
rStatPop(pop,objfunc,fitness,maxmin);

% builds a matrix storage for plotting line graphs
stats(gen,:)= [gen objbest objave chrombest];

for gen= 2:maxgen

% prints the current generation
fprintf('gen= %d (%d) %f\n',gen,maxgen-gen,objbest);

% applies reproduction
pop=
rGradSel(pop,popsize,lchrom,fitness,chrombest,fitbest,xlb,xub,etha);

% Gradient-like selection
% applies crossover
[pop,nxover]= rMsXover(pop,popsize,lchrom,pcross); % modified
simple crossover

% applies mutation
[pop,nmutat]=
rDynaMut(pop,popsize,lchrom,pmutat,xlb,xub,gen,maxgen); %dynamic
mutation

% calculates the objective function value
objfunc= EvalObj22(pop,lchrom,popsize,Test_data);

% applies Elitism
[pop,objfunc]= rElitism(pop,objfunc,chrombest,objbest,maxmin);

% applies the scaling window scheme
fitness= ScaleFit(objfunc,popsize,gam,maxmin);

% computes statistics
[chrombest,objbest,fitbest,objave,gam]=
rStatPop(pop,objfunc,fitness,maxmin);

% builds a matrix storage for plotting line graphs
stats(gen,:)= [gen objbest objave chrombest];
end

figure(1)
% plots the best and average objective function values
subplot(2,1,1)
plot(stats(:,1),stats(:,2:3))

% plots the variables of the best chromosome
subplot(2,1,2)
plot(stats(:,1),stats(:,4:lchrom+3))
axis([0 100 0 10000]);

```

```

figure(2)
Rd=273000;
Ri=10000*1.0045;
Rf=20000;
R=9;
A = 10^(107.7/20);
Gs=1000;
Kd=0.005;
Kp=1;
Ki=2;

Cf= 4.94*10^-9;
L_m=(1/8*pi*90+1/8*pi*40+2*60)*10^-3;
N=200;
mu_o=4*pi*10^-7;
X=0.0006;
S_a= 480*10^-6 *0.5;

m= chrombest(1); mu_s=chrombest(2);
I_ss=chrombest(3);L=chrombest(4);
pt1= chrombest(5);
Rs=pt1*2;

X_o=L_m/(2*mu_s);
X_1=X+X_o;

k=N^2*mu_o*S_a/4;

a=-2*k*I_ss^2/X_1^3;
b=-2*k*I_ss/X_1^2;
c=2*k*I_ss/X_1^2;

n=-Rs*b*Rf*[Rd*Cf*Kd (Rd*Cf*Kp + Kd) (Rd*Cf*Ki + Kp) Ki];

d01=Ri*Cf*Rf*m*L;
d02=Ri*Cf*Rf*m*(R + Rs) + Ri*m*Rd*Cf;
d03=-(Ri*Cf*Rf*(a*L + Rs*b*c) - Ri*m + Rd*Cf*Rs*Rf*b*Kd*Gs);
d04=-(Ri*Cf*Rf*a*(R + Rs) + Ri*a*Rd*Cf + (Rd*Cf*Rs*Rf*b*Kp +
Rs*Rf*b*Kd)*Gs);
d05=-(Ri*a + (Rd*Cf*Rs*Rf*b*Ki + Rs*Rf*b*Kp)*Gs);
d06=-Rs*Rf*b*Ki*Gs;

d=[d01 d02 d03 d04 d05 d06];

t_sample = 0:0.001:2.999;
y=step(n,d,t_sample);
plot(t_sample,Test_data(:,4)/1000,'-.',t_sample,0.21*y,'-')
axis([-0.05 0.25 0 0.00032]);
legend('Step Response','Estimated Value')
xlabel('Time[s]'),ylabel('Distance[m]')

```



```

% rInitPa22.m
%
% The RINITPA22 function initializes the parameters of a RCGA
%
% Output:
%   rseed- random seed
%   maxmin= -1 for minimization, 1 for maximization
%   maxgen- maximum generation
%   popsize- population size(must be an even integer)
%   lchrom- chromosome length
%   pcross- crossover probability
%   pmutat- mutation probability
%   xlb- lower bound of variables
%   xub- upper bound of variables
%   etha- parameter of the selection operator
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function
[rseed,maxmin,maxgen,popsize,lchrom,pcross,pmutat,xlb,xub,etha]=
rInitPa20

rseed=      937;
%rseed=input('rseed= ');
maxmin=     -1;           % -1 for minimization
maxgen=    100;
popsize=   100;           % popsize should be even
lchrom=     5;
etha=       1.7;
pcross=     0.9;
pmutat=    0.1;
xlb=  0*ones(1,lchrom);
xub= 10*ones(1,lchrom);

%xlb(1,1)=0;
%xlb(1,2)=0;
%xlb(1,3)=0.6;
xlb(1,1)=0.5;
xlb(1,2)=0;
xlb(1,3)=0.7;
xlb(1,4)=0.025;

%xub(1,1)=0.5;
%xub(1,2)=1000;
%xub(1,3)=0.8;
xub(1,1)=0.9;
xub(1,2)=10000;
xub(1,3)=0.9;

```

```

xub(1,4)=0.095;
xub(1,5)=2;

%xub(1,2)=480*10^-6;
%xub(1,2)=1*10^-3;

%xub(1,1)=10000;

%xub(1,2)=10;
%xub(1,3)=3.5*10^5;
%xub(1,3)=1*10^-7;

if(rem(popsiz, 2) ~= 0) % do not move
    popsize= popsize + 1;
end

% EvalObj22.m
%
% The EVALOBJ6 function evaluates a multivariable function
%
% Input:
%   x- variables, matrix
%   npara- number of the variables
%   popsize- population size
% Output:
%   objfunc- objective function value, vector
%
% Copyright (c) 2000 by Prof. Gang-Gyoo Jin, Korea Maritime
University
% Revision 0.9 2003/4/17
% Edit by Hwanghun Jeong, CME PKNU

function objfunc= EvalObj22(x,npara,popsiz,Test_data);
Rd=273000;
Ri=10000*1.0045;
Rf=20000;
R=9;
A = 10^(107.7/20);
Gs=1000;
Kd=0.005;
Kp=1;
Ki=2;

Cf= 4.94*10^-9;
L_m=(1/8*pi*90+1/8*pi*40+2*60)*10^-3;
N=200;
mu_o=4*pi*10^-7;
X=0.0006;
S_a= 480*10^-6 *0.5;

```

```

for i= 1:popsize
m= x(i,1); mu_s=x(i,2); I_ss=x(i,3);L=x(i,4);
pt1=x(i,5);
Rs=pt1*2;

X_o=L_m/(2*mu_s);
X_1=X+X_o;

k=N^2*mu_o*S_a/4;

a=-2*k*I_ss^2/X_1^3;
b=-2*k*I_ss/X_1^2;
c=2*k*I_ss/X_1^2;

n=-Rs*b*Rf*[Rd*Cf*Kd (Rd*Cf*Kp + Kd) (Rd*Cf*Ki + Kp) Ki];

d01=Ri*Cf*Rf*m*L;
d02=Ri*Cf*Rf*m*(R + Rs) + Ri*m*Rd*Cf;
d03=- (Ri*Cf*Rf*(a*L + Rs*b*c) - Ri*m + Rd*Cf*Rs*Rf*b*Kd*Gs);
d04=- (Ri*Cf*Rf*a*(R + Rs) + Ri*a*Rd*Cf + (Rd*Cf*Rs*Rf*b*Kp +
Rs*Rf*b*Kd)*Gs);
d05=- (Ri*a + (Rd*Cf*Rs*Rf*b*Ki + Rs*Rf*b*Kp)*Gs);
d06=-Rs*Rf*b*Ki*Gs;

d=[d01 d02 d03 d04 d05 d06];

t_sample = 0:0.001:2.999;
y=step(n,d,t_sample);
objfunc(i)=0;
for j=1:3000
obj=abs(Test_data(j,4)/1000-0.21*y(j,1));
objfunc(i)= objfunc(i)+obj;
end
end

```

Author details

Hwang Hun Jeong, So Nam Yun and Joo Ho Yang
Korea Institute of Machinery and Materials, Daejeon, Republic of Korea

Joo Ho Yang
PuKyong National University, Pusan, Republic of Korea

5. References

- [1] Lichuan Li et al. A One-Axis-Controlled Magnetic Bearing and Its Performance. JSME international Journal. Series C 2003;46(2) 391-396.
- [2] B. Polajzer et al. Decentralized PI/PD position control for active magnetic bearing . Electrical Engineering 2006; 53-59.

- [3] Hector Martin Gutierrez and Paul I. Ro. Parametric Modeling and Control of Long-Range Actuator Using Magnetic Servo Levitation. IEEE Transaction on Megnetics 1998;(34)5 3689-3695.
- [4] Susan Carlson-Skalak, Eric Maslen and Yong Teng. Magnetic Bearing Actuator Design using Genetic Algorithn. Journal of Engineering Design 1999;(10) 2 143-164.
- [5] Kuan-Yu Chen et al. A Self-tunning fuzzy PID type controller design for unbalance compensation in a active magnetic bearing. Expert System with Application 36 2009; 8560-8570
- [6] G. G. Jin. The Genetic Algorithm and the Apllications: Gyo Woo sa; 2000.
- [7] Apex. Power Optional Amplifier, Apex, pp1-4. http://www.datasheet.co.kr/datasheet-html/P/A/1/PA12_ETC.pdf.html (accessed 13 July, 2012)
- [8] Benjamin C. Kuo, The Digital Control Engineering: Hyoung Seol; 2003.